VŠB - Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra informatiky

Komplexní na webu založené řešení pro správu web hosting serverů

Complex Web-based Solution for Management of Web Hosting Servers

2010/2011 Bc. Ondřej Liška

Prohlášení

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

V Ostravě dne ....

Podpis

Abstrakt

Cílem této práce je vytvoření ovládacího prostředí pro klienty a administrátory provozující hostingové služby. Ovládacím prostředím se myslí webové rozhraní, které umožňuje klientům těchto služeb konfigurovat a měnit jejich různorodá nastavení. Dále pak možnost administrátorům spravovat základní nastavení. Tato práce je z větší části zaměřena na konfiguraci operačního systému, na kterém je aplikace provozována a také částečně na její instalaci.

Klíčová slova

Apache2 HTTP server, web server, DNS, virtual host, ftp, MySQL, php, debian server, Postfix, UNIX, linux, PowerDNS, ProFTPd, programovací jazyk C

Abstract

The goal of this thesis is to create control environment for clients and administrators running hosting services. The control environment means Web-based interface, which allows to clients configure and change varying settings of these services. Furthermore, the opportunity for administrators to manage the basic settings of services. This work is largely focused on the configuration of the operating system on which the application is operated and partly to its installing.

Keywords

Apache2 HTTP server, web server, DNS, virtual host, ftp, MySQL, php, debian server, Postfix, UNIX, linux, PowerDNS, ProFTPd, programming language C

Úvod

V roce 2011 je již internet součástí našeho každodenního života. Není pouze prostředkem pro webové aplikace, zábavu a sociální služby, ale i pro mnoho veřejných služeb, které využívají internet pro přenos vlastních dat. Mohou to být například data z bankovního sektoru nebo z různých meřících zařízení (meteorologická data). Internet se také stal místem, kde se lide, ale především firmy snaží oslovit takovou skupinu lidí, která jim přinese zisk. Proto abychom se mohli prezentovat na internetu potřebujeme tak zvanou internetovou prezentaci.

Internetovou prezentaci nebo-li www stránky, potřebujeme někde umístit tak, aby byly dostupné veřejnosti. A proto je zde velké množství společností, které nám nabízí služby v oblasti hostování našich prezentací. Doba, kdy jsme od takovéto společnosti dostali pouze přístupy k FTP účtu je již snad nenávratně pryč a nemalá část klientů by si ráda nastavila některé parametry sama. Například vytvoření FTP účtů pro různé uživatele nebo více databází.

V současnosti je v oblasti webových serverů velmi oblíbený a populární operační systém GNU/Linux. Dále pak na tomto operačním systému webový server Apache, jazyk PHP, a také databázový server MySQL. Všechny tyto součásti jsou v oblasti poskytování hostingových služeb v dnešní době celosvětovým standardem. Jedna z nejznámějších distribucí GNU/Linux používaná pro webový server je distribuce Debian.

V době psaní této práce byla aktuální verze distribuce Debian 5.0 s označením Lenny. V průběhu psaní došlo k vydání nové verze Debian 6.0 s označením Squeeze. Celá práce je vyvinuta na verzi 5.0 (Lenny). Aplikace, která je v rámci této práce vyvíjená je pojmenovaná Hostmin a toto jméno také několikrát v práci použiji.

1 Instalace GNU/Linux Debian

Pouze ve stručnosti zde popíšu instalaci serverové distribuce Debian. Instalaci provedeme jen se základním nastavením na námi zvolený hardware. Později budeme konfigurovat celý OS tak, abychom ho připravili pro běh vyvíjené aplikace Hostmin. Cílem je nainstalovat serverovou distribuci OS Debian s minimálním počtem balíků.

Pro instalaci operačního systému Debian budeme potřebovat instalační CD nebo DVD. Distribuce je také k dispozici pro instalaci ze sítě takzvaný netinstall[1], kterou lze stáhnout z internetu. Naleznete ji na oficiálních stránkách OS Debian. Zde jsou také ke stažení instalační CD a DVD, které obsahují nejrůznější balíky s dalším doplňkovým softwarem. Pro tuto práci jsem použil výše zmíněnou možnost, instalace ze sítě, netinstall. Instalace ze sítě obsahuje minimální množství softwaru pro započetí instalace a stažení zbývajících balíčku probíhá z

[1] Síťová instalace, která je k dispozici u většiny distribucí GNU/Linux

internetu. Netinstall si mnohé potřebné balíky stáhne během instalace z repositářů pro danou distribuci a verzi. Proto potřebujeme internetové připojení již během samotné instalace. Po ní, bude pokračovat samotná konfigurace systému a doinstalování potřebných balíků. Distribuce Debian není nijak zvlášť hardwarově náročná. Já jsem si pro práci pořídil menší PC s CPU Intel Atom 1.6 GHz, 2GB RAM a 80 GB HDD.

1.1 Postup instalace

Jako první se při instalaci zobrazí úvodní obrazovka. Ta nabízí výběr instalace v textovém nebo grafickém režimu. Dále pak pokročilé možnosti a nápovědu. Zvolíme možnost instalace v textovém režimu. Pokud preferujete grafický režim, můžete si vybrat ten.

Následuje zvolení jazyku, ve kterém bude probíhat instalace systému. Výchozí hodnotou je angličtina. Můžeme si vybrat češtinu, podle toho komu co vyhovuje. Dalším krokem je geografické umístění serveru. Česká Republika se nachází pod položkou `other`. Pokračujeme výběrem rozložení klávesnice. Po instalaci si můžeme další jazyky jednoduše doinstalovat. Proběhne automatické zjištění nastavení a konfigurace sítě. Ta bude nakonfigurována na dynamické získávání IP adresy. Později po instalaci konfiguraci připojení změníme.

Instalace pokračuje dotazem na doménové jméno. Jako obecný příklad můžeme použít jméno *example.com*. Já jsem při vytváření práce zaregistroval novou doménu *xliska.cz*, kterou jsem pomocí DNS záznamů nasměroval na veřejnou IP adresu přidělenou mým poskytovatelem internetu. K této IP jsem připojil také svůj malý server.

Pokračujeme rozdělením disku. Pro zjednodušení si vybereme možnost řízeného rozdělení disku nebo jinak řečeno asistovaného rozdělení disku. Vybereme disk, na který systém nainstalujeme. Na něm vytvoříme jeden velký oddíl pro systém (/) a jeden tak zvaný odkládací oddíl neboli swap. Protože se jedná o řízené rozdělení, instalace se dotáže, jestli chceme všechny data na jednom disku. Samotné rozložení je pouze na vlastním výběru. Obvykle se pro data samotných prezentací vybírá jiné umístění, než je diskový prostor pro operační systém. Následuje potvrzení výběru. Zvolíme dokončit rozdělování a zapsat všechny změny na disk.

Jakmile se dokončí proces vytváření disku, začnou se soubory operačního systému kopírovat na disk. Po skončení již zbývá jen pár kroků, ale o to mnohem více důležitých. Nejdříve je potřeba vytvořit heslo uživatele `root`, který má veškerá práva a povolení na všechny soubory a programy. Mělo by být dostatečně silné. Pro jeho vytvoření nejsou určena žádná pravidla, ale doporučuje se, aby mělo minimálně osm znaků dlouhé, kombinace číslic, malých a velkých písmen, případně nějaký speciální znak. Samozřejmostí je jeho následné potvrzení opětovným napsáním.

Používání `root` účtu se z bezpečnostního hlediska nedoporučuje. Proto je při instalaci vytvořen i jiný takzvaný běžný účet. Instalátor se zeptá na celé jméno

uživatele a následně uživatelské jméno. Můžeme ho pojmenovat například jako `Administrator`. Dalším krokem je vytvoření a potvrzení hesla pro právě vytvořený účet. Protože je použita varianta netinstall, která obsahuje jenom minimum balíčků, je potřeba vybrat pro další postup umístění repositářů. Z tohoto umístění budou stahovány další potřebné balíky pro instalaci. Můžeme si vybrat z několika zdrojů podle geografického umístění. Vybereme si český. Pokud používáme HTTP proxy, napíšeme jeho adresu. Pokud žádný proxy server nepoužíváme, necháme pole prázdné.

V následujícím bodě provedeme konfiguraci správce balíku APT. Zda se chceme zúčastnit ankety ohledně používání balíku je na vlastní volbě. Jedním z posledních kroků je výběr softwaru. Mezi volbami můžeme najít web server, print server, DNS, poštovní server a další často používané balíky. Z těchto balíku neinstalujeme žádný. Necháme označenou pouze volbu `standart system`. Všechny ostatní potřebné balíky a software se nainstalují přímo z operačního systému manuálně.

V této části instalace jsme již téměř u konce. Zbývá pouze potvrdit instalaci `GRUB LOADERU` na disk. Potvrdíme, vyjmeme instalační CD a restartujeme počítač. V případě, že se celá instalace obešla bez problému, máme úspěšně nainstalovaný operační systém Debian 5.0 (Lenny).

2 Základní nastavení

Samotná instalace Debian není dostačující. Kromě instalace dodatečného softwaru je potřeba minimální nastavení operačního systému a několik nutných balíků.

Čas od času je potřeba udělat v systému údržbu nebo jen naléhavě provést různé změny nebo opravy. Proto potřebujeme občas k serveru přistoupit vzdáleně. K zabezpečenému vzdálenému připojení je velmi často využíváno SSH, které slouží pro bezpečné připojení a přenos souborů. Debian neinstaluje ve výchozím nastavení balík OpenSSH. Tento balík nahrazuje například telnet nebo rlogin, které nemají šifrovaný přenos hesla. S OpenSSH se zároveň instaluje službu sshd, která nám umožní zmiňované vzdálené připojení k serveru. Pro funkčnost aplikace není tento balík podmínkou, ale doporučuji ho nainstalovat. Přidáme také balík pro diskové kvóty neboli omezení diskového prostoru pro uživatele a skupiny. Instalaci provedeme příkazem:

```
atp-get install ssh openssh-server openssl quota quotatool
```

Po nainstalování `openssh-server` se můžeme pomocí SSH připojit k serveru vzdáleně. Balíček `openssl` nám později poslouží pro vygenerování SSL certifikátů, které budeme potřebovat v dalších částech konfigurace.

Během instalace samotný instalátor nakonfiguroval naše internetové připojení na získávání nastavení sítě přes DHCP. Toto nastavení změníme. Server by měl mít statickou IP adresu. Pro takovou změnu musíme editovat soubor

s konfigurací sítě, který je umístěn v `/etc/network/interfaces`. V něm upravíme nastavení podle našich potřeb. Abychom po změně nemuseli restartovat celý systém, je nutné řádek `allow-hotplug eth0` změnit na `auto eth0`. Soubor by měl být podobný níže uvedenému[8].

Uvedené adresy a rozhraní `eth0` jsou pouhou ukázkou. Nastavení upravíme podle přidělených adres od našeho poskytovatele internetového připojení.

```
auto lo
iface lo inet loopback
auto eth0
iface eth0 inet static
        address 192.168.100.5
        netmask 255.255.255.0
        network 192.168.100.0
        broadcast 192.168.100.255
        gateway 192.168.100.1
```

Aby se nové nastavení načetlo, je potřeba restartovat síťové rozhraní. To uděláme příkazem:

```
/etc/init.d/networking restart
```

Dále je potřeba nastavit hostname. A to tak, že upravíme soubor `/etc/hosts`. Do tohoto souboru přidáme pod řádek s localhost adresou další řádek s naší IP adresou a hostname. Podobně jako tento[8]:

```
192.168.100.5   server.example.com    server
```

Dále spustíme příkaz [8]:

```
echo server.example.com > /etc/hostname
```

A ještě [8]:

```
/etc/init.d/hostname.sh start
```

Nakonec si ověříme, jestli se nám změna hostname podařila. Napíšeme do terminálu tyto dva příkazy. Oba by nám měli vrátit stejný výsledek[8].

```
hostname
hostname –f
```

Je důležité, aby nám oba poslední příkazy vypsaly stejný výsledek doménového jména server.example.com. První příkaz vypíše doménové jméno, druhý vypíše plně specifikované doménové jméno počítače. Jinak řečeno, doménové jméno může být pouze část plně specifikovaného doménového jména (zkratka FQDN).

Abychom mohli používat omezení diskového prostoru pro uživatele, musíme nastavit také balík `quota`. Nejdříve upravíme soubor `/etc/fstab`, do kterého za hodnotu `defaults` u konkrétního systému souborů, v tomto případě „/“ přidáme hodnoty `usrquota,grpquota`. Hodnoty oddělujeme čárkami. Tím aktivujeme kvóty na systému souborů. Poté musíme server restartovat. Předposledním krokem pro nastavení diskového omezení je vytvoření souborů kvót na systému souborů. Příkaz zadejte až po restartu serveru[9]:

```
quotacheck -F vfsold -c -a -v
quotacheck -F vfsold -c -a -g
```

Ještě nastavíme cron pro zasílání týdenního reportu o diskovém omezení příkazem:

```
crontab -e
```

A vložíme do nově otevřeného souboru řádek:

```
0 3 * * 0 /sbin/quotacheck -avug
```

První příkaz otevře konfigurační soubor cron aktuálního uživatele. Veškerou konfiguraci provádějte jako uživatel `root`. O nastavení kvót pro jednotlivé uživatele se již postará aplikace Hostmin.

Také je potřeba úprava zdrojů software. Zdroje software najdeme v souboru umístěném v `/etc/apt/source.list`. Z tohoto seznamu získává informace správce balíků APT. Pokud se v souboru vyskytuje zdroj obsahující mechaniku CD, dáme ho do komentáře. Provedeme aktualizaci databáze zdrojů příkazem `apt-get update` a nakonec aktualizaci celého systému příkazem `apt-get upgrade`.

Důležitou součástí je také synchronizace systémového času. Systémové hodiny si při startu počítače nastaví čas podle hardwarových hodin a pak se už o systémový čas stará časovač procesoru. V časovači procesoru dochází k občasným odchylkám a systémový čas přestává být přesný. To může mít vliv na některý software, který by nemusel fungovat správně. Odchylky jsou nežádoucí také pro administrátory, pro které se špatný čas zaznamenává do logovacích souborů.

Vzhledem k tomu, že se jedná o server, u kterého se předpokládá dlouhodobý běh před dalším restartem, mohlo by tak dojít k synchronizaci s hardwarovými hodinami, až za několik dnů nebo později. Proto je synchronizace systémového času prováděná přes internet. Spolehlivou variantou je instalace `ntp` a `ntpdate`[8].

```
apt-get install ntp ntpdate
```

Od této chvíle bude systém automaticky synchronizovat datum a čas pomocí NTP protokolu přes internet. Konfigurační soubor `ntp` najdeme v `/etc/ntp.conf`. V něm lze vidět již přednastavené servery pro synchronizaci

času. Pokud bychom chtěli, můžeme seznam serverů doplnit nebo nahradit o námi vybrané servery.

3 Instalace a konfigurace dodatečného softwaru

Po úspěšné instalaci serveru a jeho nabootování můžeme začít instalovat a konfigurovat další software. Protože se jedná o server, na kterém budeme prezentovat webové stránky, ale také provozovat mailový, databázový a DNS server, je potřeba nainstalovat minimálně níže uvedený software. K níže uvedeným balíkům existuje také jejich alternativa. U každého z nich je napsáno o různých výhodách a částečně i proč je použit konkrétní balík. Aplikace je však navržena a postavena na těchto konkrétních balících. Každá instalace dalšího software, který spolu výrazně nesouvisí bude popsána samostatně.

3.1 Apache HTTP server

Jedná se o HTTP server pro operační systémy UNIX i Windows NT. Aktuální verze serveru během psaní této publikace byla 2.2 a proto se často mluví pouze o apache2. K tomuto HTTP serveru existuje mnoho alternativ. Většina těchto variant je vhodná pro různé situace, stejně jako samotný Apache server. Jedna z těchto variant, která je velmi oblíbená je *lighttpd* server. Je to jednodušší verze, která je méně náročná na systémové i hardwarové prostředky a nevyužívá tolik procesorového času jako Apache. Zároveň je dostatečně rychlý a bezpečný. Stejně jako *Apache* disponuje řadou funkcí, jako je FastCGI nebo HTTP-proxy. Samozřejmostí je podpora skriptovacích jazyků PHP, RUBY, Python a další.

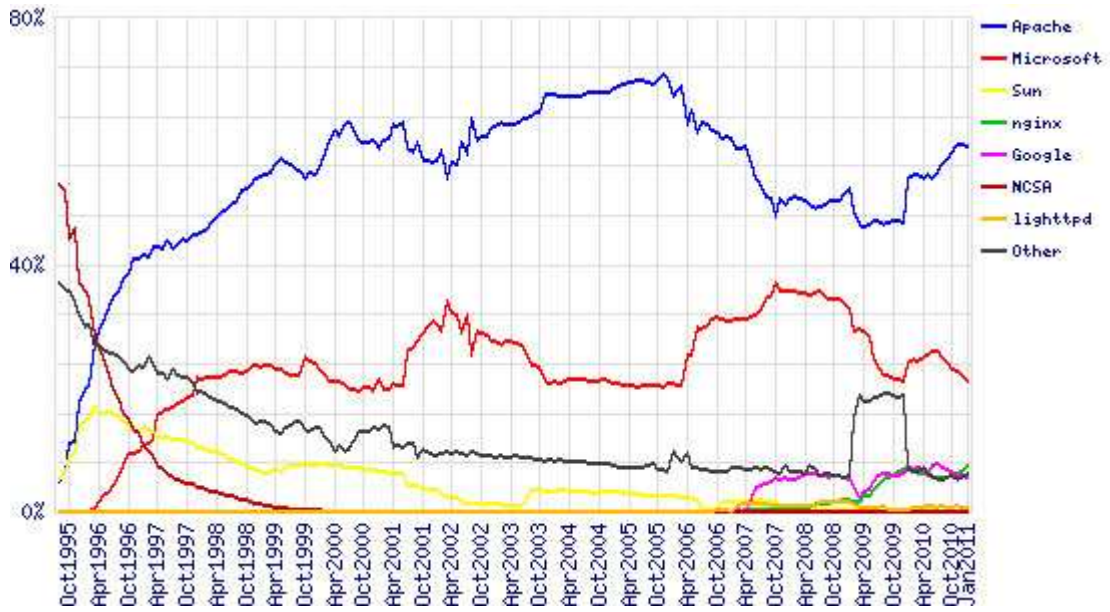
Jiným oblíbeným HTTP serverem je více komplexní NGINX. Ten zahrnuje mezi svou funkcionalitu také IMAP/POP3 proxy server. Od roku 2004, kdy byla vydána jeho první oficiální veřejná verze, si dokázal získat třetí pozici mezi nejpoužívanějšími HTTP servery. Také NGINX disponuje širokou škálou doplňků a funkcí jako je již zmiňované FastCGI, SCGI, SSL a další.

Zajímavým na poli webových serverů je GWS neboli Google web server. Tento server je používán pouze společností Google a nejsou o něm známé téměř žádné informace. I přesto zastává podle dostupných informací čtvrtou pozici mezi nejpoužívanějšími HTTP servery. Pořadí se však mění pokud se zaměříme pouze na aktivní domény. Tam Google web server předbíhá NGINX a dostává se na třetí pozici. Tím vzniká poněkud překvapivá situace, vzhledem k tomu, že o Google web serveru není známo mnoho informací.

Narozdíl od *lighttpd*, NGINX nebo GWS má Apache HTTP server nejdelší historii a také proto je světovou jedničkou a standardem pro oblast webových serverů. Podle společnosti NETCRAFT2, která se zabývá výzkumem

[2] http://www.netcraft.com

a analýzou dat v mnoha směrech Internetu, je Apache používán na více než 57% serverů. Poslední výzkum v době psaní páce byl z února 2011 viz. obr. 1.



Obrázek č. 1: Webové servery napříč doménami Červen 2000 až Únor 2011
Zdroj: http://www.netcraft.com

Přehlednější čísla a procentuální vyjádření v tab. 1.

Server	Leden 2011	v %	Únor 2011	v %	Změna
Apache	161,591,445	59.13	171,195,554	60.10	0.98
Microsoft	57,392,351	21.00	57,084,126	20.04	-0.96
nginx	20,504,634	7.50	21,570,463	7.57	0.07
Google	15,112,532	5.53	14,454,484	5.07	-0.46
lighttpd	1,866,872	0.68	1,953,966	0.69	0.00

Tabulka č. 1: Statistika webových serveru v internetu
Zdroj: http://www.netcraft.com

Výchozí instalace Apache 2 v Debian Lenny obsahuje několik modulů. Aktivovaný je pouze `mod_userdir`. Ostatní se musí aktivovat po instalaci nebo dodatečně stáhnout. Proto rovnou nainstalujeme i další moduly[3], které se nám budou hodit. Instalaci provedeme příkazem[4]:

```
apt-get install apache2 apache2.2-common apache2-doc \
apache2-mpm-prefork apache2-utils libexpat1 ssl-cert \
libapache2-mod-php5 libapache2-mod-fcgid apache2-suexec \
libapache2-mod-suphp
```

[3] Více informací o jednotlivých modulech na http://httpd.apache.org/docs/2.0/mod/

Celý příkaz zadejte najednou nebo rozdělte na několik oddělených příkazů, kdy každý začíná `apt-get install`.

Nejdříve si aktivujeme potřebné moduly. Je běžné používat modul `rewrite` pro takzvané SEO optimalizace, ten aktivujeme určitě. Dále aktivujeme modul pro `ssl`, `suexec`, `actions` a `include`. Pro aktivaci modulů se používá příkaz `a2enmod nazev_modulu` a pro deaktivaci pak `a2dismod nazev_modulu`. Pokud chceme aktivovat nebo deaktivovat více modulů najednou, oddělíme jednotlivé názvy mezerou[4]:

```
a2enmod rewirte ssl suexec actions include
```

Pro službu HTTPS je potřeba vygenerovat certifikát a nastavit apache pro naslouchání na portu 443, který je určený právě pro HTTPS. Port nastavíme v `/etc/apache2/ports.conf`. Přidáme následující řádek[4]:

```
Listen 443
```

Abychom mohli HTTPS úspěšné používat musíme ještě vygenerovat certifikát pro Apache. K tomu použijeme `openssl`, který jsme si nainstalovali v části základní nastavení. Jedná se o tak zvaný self-signed certifikát. Kdy certifikát není podepsaný žádnou certifikační autoritou, ale námi[10].

```
openssl req –new –x509 –days 365 –nodes –out \
/etc/apache2/ssl/hostmin.pem –keyout \
/etc/apache2/ssl/hostmin.pem
```

Takovýto soubor je určený vždy pro jednu doménu. Během vytvoření budeme dotázání na „Common Name (CN)“ kde doplníme název domény. Řešení se hodí pouze pro firemní sítě nebo pouze provoz aplikace Hostmin. S certifikátem ještě budeme pracovat. Po takové konfiguraci je nutné apache restartovat:

```
/etc/init.d/apache2 restart
```

Základní nastavení apache2 je hotové. Později budeme ještě do konfigurace zasahovat pro nastaveni naší aplikace Hostmin.

3.2 Databáze MySQL

Světově nejvíce oblíbená databáze je právě databáze MySQL. Během její instalace je potřeba vyplnit několik otázek, jako je hlavní neboli root uživatel databáze a jeho heslo. Instalace MySQL se provede příkazem[8]:

```
apt-get install mysql-client mysql-server
```

MySQL nám nabízí dvě možnosti svého provozu. První možnost, která je výchozí, je lokální provoz. To znamená, že přihlášení k databázi a databáze samotná, je přístupná pouze ze serveru stejného serveru, na kterém běží daný

MySQL server. Není přístupná vzdáleně. Druhá možnost je vzdálený přístup a přihlášení do databáze. Taková možnost se musí povolit a to tak, že v souboru `/etc/mysql/my.cnf` zakomentujeme řádek:

```
#bind-address   = 127.0.0.1
```

V tomto řádku „#“ označuje komentář. Pro načtení nového nastavení restartujeme mysql server:

```
/etc/init.d/mysql restart
```

Tohle je z nastavení mysql všechno. Aktuální verze MySQL je 5.5.

3.3 PHP 5 a PHPMyAdmin

Nejrozšířenějším a možná také nejpoužívanějším skriptovacím jazykem v oblasti webových aplikací je PHP. Existuje pro něj velké množství frameworku a je možné ho použít na UNIX i Windows systémech. Aktuální verze při psaní a vývoji aplikace je 5.3. Pro PHP je nutné nainstalovat modul web serveru apache2. To jsme však už udělali při jeho instalaci. Zbývá jen doinstalovat samotné PHP a s ním i podporu pro MySQL databázi. Všechny balíky nainstalujeme najednou příkazem[8]:

```
apt-get install php5 php5-common php5-gd php5-mysql php5-imap \
phpmyadmin php5-cli php5-cgi php-pear php-auth php5-mcrypt \
mcrypt php5-imagick imagemagick
```

Do instalace jsem přidal rovnou několik běžně používaných modulů PHP a také balík PHPMyAdmin. Během vývoje byla použita verze PHPMyAdminu 3.3.9. Jedná se o webovou aplikaci napsanou právě v PHP pro správu databáze MySQL. Podporuje široké množství operací s databází a ty nejvíce frekventované operace jsou podpořeny příjemným uživatelským rozhraním. Vydává se ve dvou verzích. Pro MySQL 5 a PHP 5.2 a vyšší je určen PHPMyAdmin ve verzi 3.x a pro PHP 4.x a MySQL 4.x je určena verze 2.x.

Během instalace PHPMyAdminu musíme uvést, jaký je používaný webový server. Uvedeme apache2. Po dokončení celé instalace restartujeme apache2 opět příkazem:

```
/etc/init.d/apache2 restart
```

Od této chvíle můžeme spravovat databázi přes PHPMyAdmin. Pokud máme na serveru nainstalované také grafické rozhraní, lze k PHPMyAdminu přistoupit například na adrese `http://localhost/phpmyadmin/` nebo na `http://127.0.0.1/phpmyadmin/` z webového prohlížeče. Můžeme přistoupit také vzdáleně, kde místo localhost zadáme IP adresu našeho počítače.
V případě, že by ani jeden z přístupů nefungoval, musíme zkontrolovat zda se v našem webovém adresáři, který je jako výchozí nastaven na `/var/www/` nachází symlink PHPMyAdmin. Symlink označuje symbolický odkaz, který

ukazuje na jiný soubor nebo adresář. Pokud se takový soubor s názvem `phpmyadmin` v adresáři nenachází, vytvoříme ho příkazem:

```
ln -s /usr/share/phpmyadmin phpmyadmin
```

Kde `ln -s` je příkaz pro vytvoření symlinku s názvem `phpmyadmin` směřující na adresář `/usr/share/phpmyadmin`, ve kterém je PHPMyAdmin nainstalovaný.

3.4 ProFTPd

Abychom se dostali k našemu webovému prostoru, používáme většinou klasický přístup přes FTP. Vhodným řešením pro náš server může být například ProFTPd. Jedná se o balík, který zároveň poskytuje možnost spravovat FTP účty za použití databáze MySQL. Díky tomu můžeme na jednom serveru používat tisíce FTP účtu, bez ohledu na to, jestli jsou uživatelé FTP reální uživatelé operačního systému.

Pro operační systém Debian existuje již předem nakonfigurovaný balík `proftpd` a pro správu FTP přes MySQL databází, modul `proftpd-mod-mysql`. Pro jeho instalaci tedy použijeme příkaz[8]:

```
apt-get install proftpd proftpd-mod-mysql
```

Předpokládáme již předem nainstalovanou MySQL databázi, Apache 2 server, PHP a PHPMyAdmin. Během instalace je nutné zadat v jakém módu ProFTPd poběží a zadáme `standalone`. ProFTPd může fungovat jako samostatný daemon nebo také jako součást `inetd/xinetd` daemona. Možností `standalone` jsme vybrali samostatný běh. Po instalaci musíme vytvořit nového uživatele a skupinu pro potřeby ProFTPd, které bude ve výchozím stavu při přihlášení přes FTP používat UID a GID tohoto uživatele, a to v případě, že neurčíme UID a GID jiného uživatele. Skupinu a uživatele vytvoříme příkazy[5,8]:

```
groupadd -g 2001 ftpgroup

useradd -u 2001 -s /bin/false -d /bin/null -g ftpgroup ftpuser
```

Hodnotu UID a GID 2001 je nutné dodržet. Aplikace přepokládá tuto hodnotu a má ji uloženou ve své databázi. Připomínám, že příkaz `useradd` musí být na jednom řádku.

V tuto chvíli je standardním krokem vytvoření databáze, kterou bude ProFTPd používat. To už ale není nutné, protože veškeré tabulky, které jsou pro chod potřebné, jsou již obsaženy v databázi vyvíjené aplikace.

V tuto chvíli už jen zbývá nakonfigurovat ProFTPd. Prvním krokem nastavení je upravit soubor umístěný v `/etc/proftpd/modules.conf` a povolit tři moduly. Moduly se povolují příkazem `LoadModule nazev_modulu`. V souboru

si tedy najdeme tyto moduly a dáme mimo komentáře nebo případně dopíšeme[5,8]:

```
LoadModule mod_sql.c
LoadModule mod_sql_mysql.c
LoadModule mod_quotatab_sql.c
```

Tím povolíme načtení modulu pro práci s SQL dotazy. Dále pak modul podporující práci s MySQL a na posledním řádku příkaz pro zavedení modulu, který obstarává činnost s kontrolou kvót (omezení) uložené v databázi prostřednictvím SQL dotazů.

Další soubor, který musíme upravit je umístěn `/etc/proftpd/proftpd.conf`. V tomto souboru dáme do komentáře část s modulem `quotatab.c`, tedy[8]:

```
#<IfModule mod_quotatab.c>
#QuotaEngine off
#</IfModule>
```

Důvodem je, že v předchozím souboru jsme zavedli modul, který získává informace o kvótách z databáze. To je konkrétně submodul modulu `qotatab.c`, který by se při spuštění deaktivoval. Proto je lepší celou část zakomentovat, aby nedošlo k žádnému ovlivnění zavedeného modulu. Ještě upravíme část, ve které určujeme způsob autentifikace takto[8]:

```
#Alternative authentication frameworks
#Include /etc/proftpd/ldap.conf
Include /etc/proftpd/sql.conf
```

Výše uvedená část souboru znamená, že se budou uživatelé ověřovat SQL dotazem na databázi. V souboru je možné nastavit další různé konfigurační hodnoty. Ty, ale necháme ve výchozím stavu.

Posledním krokem v nastavení ProFTPd je nastavení souboru `/etc/proftpd/sql.conf`, ve kterém jsou konkrétní SQL dotazy na databázi. Nesmíme zapomenout na řádku `SqlConnectInfo` nahradit správné heslo pro uživatele `proftpd` v MySQL. Uživatele si můžeme vytvořit již teď. Jakmile vytvoříme databázi a tabulky pro naši aplikaci, budeme se mu ještě věnovat. V souboru je pro nás podstatná část mezi značkami `<IfModule mod_sql.c>` a `</IfModule>` a změníme následující hodnoty[5].

```
SQLBackend      mysql
```

`SQLBackend` nastavuje typ používané databáze. Nastavíme `mysql`. Modul je načítaný již ve výchozím nastavení, protože jsme ho zavedli v souboru `/etc/proftpd/mods.conf`[5,8].

Zapnutí podpory SQL:

```
SQLEngine on
```

Způsob autentifikace jako uživatel nebo jako skupina:

```
SQLAuthenticate users groups
```

Autentifikace šifrovaná nebo nešifrovaná (čistý text):

```
SQLAuthTypes Crypt Plaintext
```

Nastavení připojení k DB:

```
SQLConnectionInfo hostmin@localhost proftpd HESLO
```

Zde se připojujeme k databázi hostmin na lokálním stroji (@localhost), uživatelem `proftpd`. Tato hodnota zůstane stejná. Jediné co se změní je `HESLO` uživatele `proftpd` za námi zvolené. Heslo vkládejte nešifrované. Později mu přidělíme pouze oprávnění pro práci s databází aplikace.

Další hodnotou jsou informace, které využívá ProFTPd pro práci s uživateli. Názvy jednotlivých hodnot jsou shodné s názvy atributu v tabulce s uživateli, která se nachází v databázi aplikace[5]:

```
SQLUserInfo ftpuser userid passwd uid gid homedir shell
```

Informace jsou také využívány pro skupiny, kterou jsou taktéž v databázi. Hodnoty opět odpovídají názvům atributů v dané tabulce:

```
SQLGroupInfo    ftpgroup groupname gid members
```

Následující hodnota nastavuje minimální UID a GID přihlášeného uživatele:

```
SQLMinID        500
```

Dále pak vytvoření domovského adresáře pokud neexistuje nastavíme takto:

```
CreateHome on
```

Potřebujeme také nastavit zaznamenávání každého přihlášení uživatele. Zapíšeme do souboru:

```
SQLLog PASS updatecount
SQLNamedQuery updatecount UPDATE "count=count+1, \
accessed=now() WHERE userid='%u'" ftpuser
```

A také informace pokaždé když uživatel něco nahraje nebo smaže pro sledování jeho aktivity:

```
SQLLog  STOR,DELE modified
SQLNamedQuery modified UPDATE "modified=now() \
WHERE userid='%u'" ftpuser
```

Zbývá do souboru zapsat informace jak budeme pracovat s kvóty. Základem je jejich povolení:

```
QuotaEngine on
```

Započítávaní do kvót operace s adresáři (vytvářeni, mazání a podobně)

```
QuotaDirectoryTally on
```

V jakém formátu bude uživateli zobrazena informace o kvótách. Tento záznam nemá vliv na to jak jsou informace o kvótě zaznamenávaný v databázi. Možné hodnoty jsou „b“, „Kb“, „Mb“, „Gb“

```
QuotaDisplayUnits Mb
```

Následující informace popisují samostatné SQL příkazy a informace o tabulkách[5]:

```
SQLNamedQuery get-quota-limit SELECT "name, quota_type,
per_session, limit_type, bytes_in_avail, bytes_out_avail,
bytes_xfer_avail, files_in_avail, files_out_avail,
files_xfer_avail FROM ftpquotalimits WHERE name = '%{0}' AND
quota_type = '%{1}'"

SQLNamedQuery get-quota-tally SELECT "name, quota_type,
bytes_in_used, bytes_out_used, bytes_xfer_used, files_in_used,
files_out_used, files_xfer_used FROM ftpquotatallies WHERE name
= '%{0}' AND quota_type = '%{1}'"

SQLNamedQuery update-quota-tally UPDATE "bytes_in_used =
bytes_in_used + %{0}, bytes_out_used = bytes_out_used + %{1},
bytes_xfer_used = bytes_xfer_used + %{2}, files_in_used =
files_in_used + %{3}, files_out_used = files_out_used + %{4},
files_xfer_used = files_xfer_used + %{5} WHERE name = '%{6}' AND
quota_type = '%{7}'" ftpquotatallies

SQLNamedQuery insert-quota-tally INSERT "%{0}, %{1}, %{2}, %{3},
%{4}, %{5}, %{6}, %{7}" ftpquotatallies

QuotaLimitTable sql:/get-quota-limit
QuotaTallyTable sql:/get-quota-tally/update-quota-tally/\
insert- quota-tally
```

Posledním nastavením jsou bezpečnostní nastavení. Vypneme možnost přihlášení uživatele jako root uživatel:

```
RootLogin off
```

Tímto jsme dokončili nastavení souboru `sql.conf`. Soubor uložíme a restartujeme ProFTPd pro načtení nové konfigurace:

```
/etc/init.d/proftpd restart
```

To je z konfigurace ProFTPd prozatím všechno.

3.5 Postfix, Courier Mail server

V současnosti je již standardem, že je v rámci služeb hostování webových stránek k dispozici i poštovní server. Podle statistiky na webu Mail Radar[4] je nejpoužívanějším type poštovních serveru sendmail, který je hned následovaný Postfixem. Sendmail bohužel nemá nativní podporu pro virtuální uživatele v databázi MySQL. S MySQL je Postfix na tom mnohem lépe. Postfix nainstalujeme a nakonfigurujeme s podporou pro zabezpečený přenos TLS a SMTP autorizaci. Současně s tím nakonfigurujeme zabezpečenou verzi SMTP(s). Pro ověřování klientů na serveru použijeme metodu Cyrus SASL dále jen SASL. Tento balík pro ověřování lze také nainstalovat s podporou MySQL. Instalaci Postfixu s modulem pro MySQL a balíkem SASL pro ověřování uživatelů na serveru provedeme příkazem[8]:

```
apt-get install postfix postfix-mysql postfix-doc postfix-tls \
libsasl2-2 libsasl2-modules libsasl2-modules-sql sasl2-bin \
libpam-mysql
```

Během instalace je potřeba odpovědět na několik otázek. Instalace probíhá v anglickém jazyce. První otázka instalátoru je jaký typ bude výchozí instalace poštovního serveru. Zvolíme možnost `Internet Site`. Další otázkou je doménové jméno serveru. Již v základní konfiguraci jsme nastavili doménové jméno na `server.example.com`, které uvedeme i v tomto případě. Doménové jméno `server.example.com` slouží pouze jako příklad pro konfiguraci. Instalace dál pokračuje samostatně.

Postfix nepodporuje práci s kvótami. Kvóta je omezení diskového prostoru například pro uživatele nebo pro emailovou schránku. Existuje však několik doplňků, které Postfixu umožní omezovat diskový prostor pro jednotlivé schránky. Použijeme takzvanou VDA záplatu, která umožní Postfixu nastavení kvót. Nejdříve je nutné zjistit si verzi Postfixu. To uděláme příkazem[8]:

```
postconf -d | grep mail_version
```

Výsledek příkazu jsou dva řádky, kde první z nich obsahuje verzi Postfixu:

```
mail_version = 2.5.5
```

Verze samozřejmě záleží na konkrétní instalaci serveru. Pokud se tato verze liší, musíme si dohledat VDA záplatu přímo pro naší verzi. Pro aplikování VDA záplaty potřebujeme zdrojové kódy Postfixu a stáhnout také samotnou záplatu. Přejdeme do složky `/usr/src` a stáhneme zdrojové kódy příkazem[8]:

[4] http://www.mailradar.com

```
apt-get source postfix
```

Dále stáhneme samotnou záplatu. Stránky najdeme na adrese *http://vda.sourceforge.net*, kde si také můžeme zkontrolovat zda existuje záplata pro naší verzi Postfixu. Záplatu stáhneme:

```
wget \
http://vda.sourceforge.net/VDA/postfix-2.5.5-vda-ng.patch.gz
```

Rozbalíme, aplikujeme a vytvoříme nové instalační balíčky Postfixu. To vše provedeme postupným zadáváním příkazů[8]:

```
gunzip postfix-2.5.5-vda-ng.patch.gz
cd postfix-2.5.5
patch -p1 < ../postfix-2.5.5-vda-ng.patch
dpkg-buildpackage
```

Během vytváření nových balíčků, po zadání příkazu `dpkg-buildpackage`, se může zobrazit informace:

```
dpkg-buildpackage: warning: Failed to sign .dsc and .changes \
file
```

Jedná se pouze o varování a proto můžeme pokračovat. Přejdeme do nadřazeného adresáře `/usr/src`, kde už jsou nově vytvořené balíčky. Pro kontrolu můžeme vypsat adresář příkazem `ls -l` a nainstalujeme záplatu Postfixu a modulu pro MySQL příkazem:

```
dpkg -i postfix_2.5.5-1.1_i386.deb postfix-mysql_2.5.5-
1.1_i386.deb
```

Tato základní instalace není dostačující a proto bude další konfigurace probíhat pomocí speciálního příkazu `postconf -e`. Příkaz se používá ve tvaru[3,6]:

```
postconf -e "parametr = hodnota"
```

3.5.1 Konfigurace postfixu

Veškeré zde nastavované hodnoty příkazem `postconf -e` se zapisují do konfiguračního souboru `/etc/postfix/main.cf`. Proto je zde také možnost konfiguraci upravovat přímo v souboru. Tento způsob konfigurace může být pohodlnější.

Nejprve nastavíme Postfix pro doručování lokální pošty. To uděláme nastavením parametru `mydestination`. Parametr může obsahovat i více hodnot, které se oddělují čárkami. Typickým nastavením je *hostname* v našem případě `server.example.com` a `localhost`. Pro zařazení celé `example.com` musíme tuto doménu buď dopsat nebo lépe později vytvořit jako virtuální doménu. Nastavení provedeme příkazem[3]:

```
postconf -e "mydestination = server.example.com, \
localhost.localdomain, localhost"
```

Nastavíme parametr `message_size_limit` pro maximální velikost zprávy na hodnotu cca 30 MB. Pokud by jsme tento parametr nezměnili, výchozí hodnota je nastavena na cca 10 MB a zadává se v bajtech:

```
postconf -e "message_size_limit = 30720000"
```

Pokračovat budeme nastavením zabezpečení SMTP pro TLS a SSL. Nejdříve nastavíme Postfix pro používání TLS[3]:

```
postconf -e "smtp_use_tls = yes"
postconf -e "smtpd_use_tls = yes"
```

Musíme vygenerovat pro TLS certifikáty. Postačí nám self-signed certifikáty, to jsou takové, které nejsou podepsané žádnou certifikační autoritou, ale podepsány sami sebou. Certifikáty se musí importovat u klienta a proto je takové řešení vhodné pro firemní sítě. U veřejných serverů by byl import certifikátů příliš nepraktický. U veřejného serveru je tedy nutné nechat si certifikát podepsat nějakou veřejnou oficiální certifikační autoritou. Nejdříve si vytvoříme certifikát certifikační autority. Balík OpenSSL obsahuje skript pro jednodušší vytváření těchto certifikátů. Skript spustíme příkazem[3,10]:

```
/usr/lib/ssl/misc/CA.pl -newca
```

Postupujeme podle pokynů skriptu. Zadáváme informace jako je heslo, ověření hesla, název země a podobně. U položky „Common Name (CN)“ zadáme název serveru, tedy v našem případě jako příklad uváděné `server.example.com`. Skript si vytvoří adresář `/demoCA/` do kterého si uloží adresáře a soubory které potřebuje pro svou další činnost a také soukromý a veřejný klíč certifikační autority. V tuto chvíli můžeme distribuovat certifikát certifikační autority `/demoCA/cacert.pem` do klientských počítačů. Pro počítače se systémem Windows je ještě nutné provést konverzi do jiného formátu. Tu uděláme příkazem[3,10]:

```
openssl x509 -in cacert.pem -out cacert.der -outform DER
```

Ve chvíli kdy máme nainstalovaný certifikát certifikační autority na klientských zařízeních můžeme přistoupit k vytvoření serverového certifikátu. Nejprve vytvoříme požadavek na certifikát s podpisem certifikační autority. Jednotlivé volby příkazů jsou popsány v manuálové stránce OpenSSL[3,10]:

```
openssl req -new -nodes -keyout poprkey.pem -out poprkey.pem \
-days 3650
```

Opět doplňujeme informace o certifikátu a pro položku „Common Name (CN)“ doplníme název serveru, v našem případě jako příklad opět `server.example.com`.

Posledním krokem je podepsání serverového certifikátu certifikační autoritou. Podepsání a vytvoření veřejného klíče provedeme příkazem[3,10]:

```
openssl ca -policy policy_anything -out /demoCA/popubcert.pem \
-infiles /demoCA/poprkey.pem
```

Během podepisování musíme zadat heslo, které jsme zadali při vytváření certifikátu certifikační autority a potvrdit podepsání certifikátu. Zkontrolujte si také cestu k soukromému klíči `poprkey.pem`.

Zbývá jen zkopírovat certifikáty do adresáře Postfixu a zabezpečit před ostatními uživateli[3]:

```
mkdir /etc/postfix/certs/
cp /demoCA/cacert.pem /etc/postfix/certs
cp /demoCA/poprkey.pem /etc/postfix/certs
cp /demoCA/popubkey.pem /etc/postfix/certs
chown root:root /etc/postfix/certs/
chmod 600 /etc/postfix/certs/poprkey.pem
chmod 644 /etc/postfix/certs/popubkey.pem
chmod 644 /etc/postfix/certs/cacert.pem
```

Samozřejmě musíme sdělit Postfixu cestu k těmto certifikátům[3]:

```
postconf -e "smtpd_tls_key_file =
/etc/postfix/certs/poprkey.key"
postconf -e "smtpd_tls_cert_file =
/etc/postfix/certs/popubkey.pem"
postconf -e "smtpd_tls_CAfile = /etc/postfix/certs/cacert.pem"
```

Abychom nám na portu 465 naslouchala zabezpečená verze SMTP s SSL musíme udělat ještě jednu úpravu. V souboru `/etc/postfix/master.cf` musíme povolit (odkomentovat) tyto řádky[3]:

```
smtps     inet  n       -       y       -       -       smtpd
  -o smtpd_tls_wrappermode=yes
  -o smtpd_sasl_auth_enable=yes
  -o smtpd_client_restrictions=permit_sasl_authenticated,reject
  -o broken_sasl_auth_clients=yes
```

Jakmile restartujeme Postfix, na portu 465 bude naslouchat zabezpečená verze SMTP s SSL certifikátem.

Protože budeme na našem serveru provozovat několik domén, znamená to, že budeme potřebovat aby také Postfix přijímal poštu pro více domén. Současné nastavení by tak fungovalo pouze pro doménu uvedenou v parametru `mydestination`. Postfix dokáže zpracovávat zprávy pro více než jednu doménu. K tomu mu slouží dvě metody. První je nazvaná virtuální aliasové domény, druhá pak virtuální schránkové domény. U první metody jde jen

o zvětšení počtu domén, pro které je server schopný přijímat zprávy. Druhá metoda, kterou v tomto případě použijeme, je na tom lépe. Ta zvětšuje množinu domén a nepotřebuje k tomu lokální účty.

Virtuální schránkové domény používají takzvané mapy. Pro nás je podstatné, že tyto mapy umí Postfix získávat také z databáze. Oproti souborovým mapám, má několik výhod. Změny v databázi se projevují hned a nemusí se proto znovu zavádět Postfix. Na druhou stranu je vyhledávání v nich o něco pomalejší. V případě, že by jsme měli výrazný pokles výkonu, mohlo by se jednat právě o pomalé vyhledávání v databázi. Bylo by pak vhodné pro tyto mapy vytvořit vlastní databázový server.

Aplikace, která je vyvíjena v této práci, již ve své databázi obsahuje tabulky pro mapy virtuální doménové schránky. Musíme tedy jen nakonfigurovat Postfix tak, aby měl k těmto tabulkám přístup. Podpora postfixu pro MySQL už byla nainstalována na začátku. Nejdříve vytvoříme soubory s SQL dotazy na databázi. Přejdeme do adresáře `/etc/postfix` a vytvoříme první soubor s názvem `mysql-virtual_domains.cf` a do tohoto souboru vložíme následující řádky[3,6,8]:

```
user = postfix
password = HESLO
dbname = hostmin
query = SELECT domain AS virtual FROM mail_virtual_domains \
WHERE domain='%s'
hosts = 127.0.0.1
```

První řádek je jméno databázového uživatele, kterým se bude Postfix připojovat k naší MySQL databázi. Druhé je heslo, které se zadává v nešifrovaném tvaru. Tohoto uživatele a jeho heslo si vytvoříme již teď. Později mu přidělíme pouze práva pro přístup k jednotlivým tabulkám v databázi hostmin. Vytvoření můžete provést již přes nainstalovaný PHPMyAdmin. Parametr `dbname` označuje databázi, ke které se má Postfix připojit. Tam ponecháme hodnotu `hostmin`. SQL dotaz v parametru `query`, zjišťuje zda se tato doména nachází v tabulce `mail_virtual_domains`. Tato tabulka totiž obsahuje veškeré domény, pro které má Postfix zpracovávat poštu. Na posledním řádku začínající `hosts` nastavíme hostitele databáze. Protože nám databáze běží na stejném serveru jako poštovní server, nastavíme hodnotu na IP adresu `127.0.0.1`. Adresa funguje jako lokální adresa pro stejný počítač. Nyní musíme sdělit postfixu, aby používal MySQL a kde najde parametry pro virtuální doménové schránky. To provedeme nastavením mapy v hlavním konfiguračním souboru Postfixu `main.cf`, a to obdobně jako doposud:

```
postconf -e "virtual_mailbox_domains = \
proxy:mysql:/etc/postfix/mysql-virtual_domains.cf"
```

Asi nejdůležitějším je vytvoření souboru s dotazem na virtuální uživatelské schránky. Vytvoříme tedy další podobný soubor s názvem `mysql-virtual_mailboxes.cf` a přidáme do něj tyto parametry:

```
user = postfix
password = HESLO
dbname = hostmin
query = SELECT maildir FROM mail_virtual_users WHERE \
email='%s' AND active = '1'
hosts = 127.0.0.1
```

Zde je v parametru `query` dotaz na virtuální uživatelské schránky s podmínkou email, kde se při příchozí zprávě vloží místo znaků `%s` emailová adresa příjemce. V podmínce je také uvedeno `active='1'`. To se nám muže hodit, pokud chceme schránku jenom dočasně pozastavit, ne však ji úplně zrušit. Pokud by takový dotaz byl bez výsledku, zpráva bude Postfixem automaticky zahozena. Parametrem `virtual_mailbox_maps` řekneme Postfixu, kde má hledat příjemce k virtuálním schránkám:

```
postconf -e "virtual_mailbox_maps = \
proxy:mysql:/etc/postfix/mysql-virtual_mailboxes.cf"
```

Vytvoření dotazu na virtuální alias je podobné dotazu na virtuální schránky. Do souboru s názvem `mysql-virtual_mailbox_limit_maps.cf` vložíme:

```
user = postfix
password = HESLO
dbname = hostmin
query = SELECT quota FROM mail_virtual_users WHERE email='%s'
hosts = 127.0.0.1
```

V tabulce virtuálních uživatelů je uvedena také kvóta pro každou jednotlivou schránku. Postfix se pak již postará o nepřekročení těchto kvót. Parametrem `virtual_mailbox_limit_maps` řekneme Postfixu kde má hledat informace o omezení diskového prostoru pro jednotlivé schránky:

```
postconf -e "virtual_mailbox_limit_maps = \
proxy:mysql:/etc/postfix/mysql-virtual_mailbox_limit_maps.cf"
```

Nesmíme zapomenout na oprávnění vytvořených souborů. Ukládáme v nich heslo pro přihlášení do databáze v nešifrovaném tvaru. Nastavíme tedy těmto souborům vlastníka `root` s právy čtení, zápis a skupinu `postfix` s právy pro čtení. Můžeme použít i zkrácený zápis pro změnu oprávnění pomocí „*“, protože všechny soubory začínají prefixem *mysql*:

```
chown root:postfix mysql*.cf
chmod 640 mysql*.cf
```

Ke kvótám přidáme ještě pár dalších nastavení Postfixu[3,6]:

```
postconf -e "virtual_mailbox_limit_override = yes"
```

Parametr `virtual_mailbox_limit_override` nám dovoluje nastavit kvótu pro virtuální schránku menší, než je maximální velikost zprávy. Maximální velikost

zprávy jsme nastavili na začátku konfigurace Postfixu parametrem `message_size_limit` na hodnotu cca 30 MB. Při naplnění schránky by bylo také vhodné upozornit ty, kteří se snaží doručit nějakou zprávu. Text upozornění můžeme nastavit parametrem[3,6]:

```
postconf -e "virtual_maildir_limit_message = "The user you are \
trying to reach is over quota.""
```

Můžeme také nastavit chování při doručení zprávy do plné schránky. Výchozí chování je nastaveno na přijetí zprávy do fronty a vrácení informace o dočasné chybě s kódem 4xx. Takováto zpráva bude doručena hned jak bude uvolněno místo pro doručení. Parametr `virtual_overquota_bounce` nastaví chování na vrácení zprávy s kódem 5xx jejímu odesílateli a zpráva bude zahozena[3,6]:

```
postconf -e "virtual_overquota_bounce = yes"
```

Nakonec konfigurace virtuálních schránek musíme ještě specifikovat, které UID a GID bude používat doručovací agent `virtual`, ve chvíli kdy bude zapisovat zprávu do schránky příjemce. Existuje více možnosti. Vytvoření mapy pro MySQL databázi, klasické soubory nebo použít staticky jednoho systémového uživatele. V našem případě využijeme poslední možnost a proto si vytvoříme nového uživatele pro doručovacího agenta[3,6]:

```
useradd -u 5000 vmail -m
```

Příkaz automaticky vytvoří uživatel `vmail` s UID `5000`, skupinu `vmail` s GID `5000` a jeho domovský adresář `/home/vmail`. Do této složky budou vytvářeny poštovní schránky. Nejdříve sdělíme Postfixu, které UID a GID použít pro doručovacího agenta[3,6]:

```
postconf -e " virtual_uid_maps = static:5000"
postconf -e " virtual_gid_maps = static:5000"
```

A také musíme sdělit Postfixu místo pro poštovní schránky virtuálních uživatel:

```
postconf -e "virtual_mailbox_base = /home/vmail"
```

Základní konfigurace Postfixu je hotová, ale ještě se musíme zaměřit na další část jeho zabezpečení. Nechceme aby byl server otevřený a poštu mohl skrze něj posílat kdokoliv. Takový server je často využívaný pro rozesílání nevyžádané pošty. Existuje několik způsoby autentifikace. Například omezení na IP adresy odesílatele. V současnosti jsou již běžné mobilní zařízení a takovéto omezení na IP adresy by bylo příliš striktní a vázané na jedno místo. Dalším možným řešením je VPN nebo použití certifikátu. Pro nás bude nejvhodnější řešení autorizace podle jména a hesla. Proto nastavíme Postfix tak, aby využíval autentizační metodu SASL následujícími příkazy.

Aktivace autentizace na straně serveru a ověřování metodou SASL[3]:

```
postconf -e "smtpd_sasl_auth_enable = yes"
```

Pro některé nestandardní klienty jako jsou starší verze MS Outlook, MS Outlook Express a další je nutné ještě nastavit parametr `broken_sasl_auth_clients`. Důvodem je nabízení SMTP AUTH podle standardu RFC 2222 a nestandardní klienti ji nerozumí. Očekávají jiný zápis.

```
postconf -e "broken_sasl_auth_clients = yes"
```

Parametrem `smtpd_recipient_restrictions` sdělíme Postfixu jaká jsou omezení pro příjemce. Jeho hodnoty jsou nastaveny na příjemce uvedené v `mynetworks` a také umožní ověřeným uživatelům odeslat email do jiných destinací[3].

```
postconf -e "smtpd_recipient_restrictions = permit_mynetworks, \
permit_sasl_authenticated"
```

Důležitým nastavením je definování ověřovacího mechanismu nabízeného klientům Postfixu. Hodnota `noanonymous` zajistí, aby server skutečně ověřil klientovy autentizační údaje[3].

```
postconf -e "smtpd_sasl_security_options = noanonymous"
```

Musíme také nakonfigurovat samotný SASL. Ten nabízí dvě autentizační služby. První se jmenuje saslauthd, která je samostatný daemon a muže být spouštěna jen s oprávněním uživatele root. Autentizaci lze provádět pouze k procedurám, které vyžadují právě oprávnění `root` (např. `/etc/shadow`). Druhá služba se jmenuje auxprop. Dokáže ověřovat hesla s různými pomocnými moduly. Každý je určený pro něco jiného, například moduly sasldb2 nebo sql.

Pro naši konfiguraci použijeme službu saslauthd. Je kompatibilní s ověřovacím mechanismem PAM (Pluggable Authentication Motdules). A modul PAM je kompatibilní s databázi MySQL. Celý tento balík jsme si nainstalovali již na začátku. Saslauthd umí pracovat pouze s ověřovacím mechanismem PLAIN a LOGIN.

Konfiguraci provedeme vytvořením souboru `/etc/postfix/sasl/smtpd.conf` a do souboru vložíme konfigurační parametry. Nejdříve nastavíme úroveň zaznamenávání chyb, varování a dalších zpráv[3,6]:

```
log_level: 3
```

Jednotlivé úrovně mohou nabývat hodnot uvedených v tab. 2.

Hodnota	Popis
0	Žádná hlášení
1 (Výchozí)	Neobvyklé chyby
2	Všechny chyby při autentizaci
3	Varování
4	Více než úroveň 3
5	Více než úroveň 4
6	Sledování interních protokolů
7	Sledování interních protokolů včetně hesel

Tabulka č. 2: Hodnoty úrovně hlášení
Zdroj: /usr/include/sasl/sasl.h

Dále musíme nastavit službu pro ověřování hesla. Jak už bylo zmíněno, nastavíme službu saslauthd:

```
pwcheck_method: saslauthd
```

Pokračujeme nastavením ověřovacího mechanismu. Využijeme oba dva. PLAIN i LOGIN.

```
mech_list: PLAIN LOGIN
```

Ke správné funkčnosti musíme udělat ještě pár úprav. Upravíme soubor `/etc/default/saslauthd`. Na řádku `START` odstraníme komentář (#) a změníme hodnotu na:

```
START=yes
```

Stále ve stejném souboru změníme parametry `mechanisms` a `options`:

```
MECHANISMS=“pam“
...
OPTIONS="-c -m /var/spool/postfix/var/run/saslauthd -r"
```

Dalším krokem je vytvoření adresáře, protože nejspíš nebude existovat:

```
mkdir -p /var/spool/postfix/var/run/saslauthd
```

Je to z toho důvodu, že nám Postfix běží odděleně. Proto musíme připravit na oddělený běh i saslauthd. Nakonec musíme přidat do skupiny sasl také uživatele postfix:

```
adduser postfix sasl
```

Přístup k procedurám podporovaným PAM zajistíme vytvořením konfiguračního souboru `/etc/pam.d/smtp`. Soubor se musí jmenovat `smtp`. Je to dáno domluveným formátem podle RFC 2254. Vytvoříme daný soubor a zapíše konfigurační parametry[3]:

```
auth   required   pam_mysql.so user=postfix passwd=HESLO \
host=127.0.0.1 db=hostmin table=mail_virtual_users \
usercolumn=email  passwdcolumn=password cryp=1

account sufficient pam_mysql.so user=postfix  passwd=HESLO \
host=127.0.0.1 db=hostmin table=mail_virtual_users \
usercolumn=email passwdcolumn=password crypt=1
```

Konfigurace sděluje PAM, že autentizace proběhne s využitím modulu `pam_mysql.so`. Tento modul potřebuje další parametry pro přihlášení do MySQL databáze. Uživatel se definuje parametrem `user` a je nastaven na `postfix`. Heslo se vkládá nešifrované do parametru `passwd`. Uživatele jsme si vytvořili už v předcházejícím kroku. Hostitelem je lokální databáze, proto také uvedená IP adresa. Databáze je `hostmin`. Později přidělím uživateli postfix oprávnění pro práci s konkrétními tabulkami v databázi `hostmin`. Tabulka pro autentizaci má název `mail_virtual_users`. Parametr `usercolumn` a `passwdcolumn` říká PAM, které atributy v tabulce odpovídají uživatelskému jménu a heslu, kterými se bude klient autentizovat. Jako uživatelské jméno bude sloužit celá emailová adresa. Posledním parametrem je `crypt`, který sděluje, v jakém formátu je heslo uloženo. Všechny ostatní parametry necháme tak, jak je uvedeno výše, pouze změníme hodnotu `HESLO` na správné, námi zvolené heslo. Nastavení PAM je pouze pro tuto práci. Existuje zde další spousta konfiguračních možností, které však nejsou pro aplikaci Hostmin důležité.

3.5.2 Konfigurace Courier Mail server

Ke správnému poštovnímu serveru patří také služby POP3 a IMAP pro výběr pošty. Pro provozování těchto služeb na našem serveru se výborně hodí balík Courier Mail Server. Jeho konfigurace je mnohem jednoduší a kratší než konfigurace Postfixu. Instalace obsahuje více balíků najednou[8]:

```
apt-get install courier-authdaemon courier-authlib-mysql \
courier-pop courier-pop-ssl courier-imap courier-imap-ssl \
courier-maildrop
```

Během instalaci Courier automaticky vytvořil SSL certifikáty pro POP3-SSL a IMAP-SSL služby. Ty obsahují špatné údaje, proto musíme vytvořit vlastní certifikáty, které již budou obsahovat správné údaje. Přejdeme do složky `/etc/courier/` a smažeme certifikáty[3]:

```
rm -f /etc/courier/imapd.pem
rm -f /etc/courier/pop3d.pem
```

Před vygenerováním nových certifikátů musíme změnit údaje ve dvou souborech. Nejprve editujeme soubor `/etc/courier/pop3d.cnf` a změníme hodnotu CN na `server.example.com`. To stejné uděláme v souboru `/etc/courier/pop3d.cnf`. Ostatní hodnoty v souborech můžeme změnit také, pokud je to nutné. Certifikáty pak znovu vytvoříme příkazy[8]:

```
mkimapdcert
mkpop3dcert
```

Dalším krokem konfigurace Courieru je nastavení přístupu do MySQL pro autorizace uživatelů. Údajů pro ověření je několik. Výhodou je, že tabulka používaná pro ověřování v Postfixu pro SMTP je vhodná i pro ověřování uživatelů Courier. Pro nastavení přístupu databáze a vytvoření SQL dotazu musíme editovat soubor `/etc/courier/authmysqlrc`. Přihlašovací údaje použijeme stejné jako pro Postfix. Problém ale nebude ani ve chvíli kdy použijete jiného uživatele. Pokud by jsme ho použili, musíme mu udělit přístup do databáze aplikace Hostmin. Heslo je opět zadáno v nešifrovaném podobě[3,8]:

```
MYSQL_SERVER 127.0.0.1
MYSQL_USERNAME postfix
MYSQL_PASSWORD postfix-heslo
MYSQL_PORT 0
MYSQL_DATABASE hostmin
MYSQL_USER_TABLE mail_virtual_users
MYSQL_CRYPT_PWFIELD password
MYSQL_UID_FIELD 5000
MYSQL_GID_FIELD 5000
MYSQL_LOGIN_FIELD email
MYSQL_HOME_FIELD „/home/vmail“
MYSQL_MAILDIR_FIELD maildir
MYSQL_QUOTA_FIELD quota
```

Databáze je nastavena opět na `hostmin`. UID a GID musíme nastavit na statické hodnoty `5000`. Hodnoty totiž odpovídají vytvořenému uživateli `vmail` a pouze tento uživatel má přistup do adresářů s poštou. Jako přihlašovací údaj nám bude sloužit celá emailová adresa a samozřejmě heslo. Adresář s poštou je umístěn v `/home/vmail`, to se shoduje s nastavením Postfixu. Atribute `maildir` označuje v databázi název adresáře pro konkrétní emailovou adresu ve kterém jsou uloženy emailové zprávy (ve tvaru maildir). Poslední řádek sděluje Courieru, kde se nachází informace o kvótě.

Nakonec nám zbývá poslední konfigurační soubor `/etc/courier/authdaemonrc`. V něm musíme změnit název modulu, který bude Courier používat pro ověřování uživatelů. Musíme změnit řádek s parametrem `authmodulelist` na[3,8]:

```
authmodulelist="authmysql"
```

Restartujeme všechny služby Courieru a Postfixu pro načtení změn:

/etc/init.d/courier-authdaemon restart
/etc/init.d/courier-imap restart
/etc/init.d/courier-imap-ssl restart

/etc/init.d/courier-pop restart
/etc/init.d/courier-pop-ssl restart
/etc/init.d/postfix restart
/etc/init.d/saslauthd restart

Konfigurace poštovního serveru je dokončena. Ještě si můžeme otestovat zda nám služby POP3 a IMAP fungují například přes telnet. Zadáme příkaz:

```
telnet localhost pop3
```

nebo

```
telnet localhost imap
```

V obou případech bychom měli dostat ve výpisu `OK`.

3.6 PowerDNS

PowerDNS je dynamický jmenný server. Umožňuje provozovat primární, sekundární DNS nebo lze použít jako cache server. Jeho hlavní výhodu je, že veškeré své záznamy ukládá do databáze. Variantou může být ukládání do souborů. Ukládání do databáze má však řadu výhod. Pro PowerDNS existuje několik webových rozhraní pro správu DNS záznamů nebo si lze vytvořit vlastní. Vlastní rozhraní je vytvořeno v rámci této diplomové práce a je součástí aplikace Hostmin. Aktuální verze v době psaní této práce byla 2.9.21. Konfigurace není vůbec náročná. Instalaci provedeme příkazem[8]:

```
apt-get install pdns-server pdns-backend-mysql
```

V tuto chvíli již nemusíme vytvářet databázi a tabulky pro naše DNS záznamy. Všechny potřebné tabulky jsou již připravené v databázi vyvíjené aplikace Hostmin. Zbývá tedy nakonfigurovat PowerDNS tak, aby měl k databázi a tabulkám aplikace přístup. Jako první musíme PowerDNS sdělit, že budeme pracovat s MySQL databází. Proto upravíme parametr `launch` v konfiguračním souboru `/etc/powerdns/pdns.conf`[7]:

```
launch=gmysql
```

Další a zároveň poslední nutnou úpravou PowerDNS je nastavení přístupových údajů k databázi. Proto si vytvoříme uživatele s názvem `powerdns` v MySQL databázi. Pro vytvoření můžeme opět použít webového rozhraní PHPMyAdmin. Konkrétní oprávnění mu přiřadíme později. Poté nastavíme přístupy v souboru `/etc/powerdns/pdns.d/pdns.local`[7]:

```
gmysql-host=127.0.0.1
gmysql-user=powerdns
gmysql-password=HESLO
gmysql-dbname=hostmin
```

Heslo se opět zadává v nešifrovaném tvaru. Z konfigurace PowerDNS je to vše. Restartujeme PowerDNS server:

/etc/init.d/pdns restart

V tuto chvíli je PowerDNS připraven a po nainstalování Hostminu bude plně funkční.

3.7 AWStats

Posledním balíkem, který potřebujeme pro plnou funkčnost aplikace Hostmin je balík AWStats. Je to jednoduchý nástroj pro analýzu logovacích souborů a je také velmi oblíbený. Instalace se provede obvyklým způsobem[8]:

```
apt-get install awstats
```

Konfigurační soubory pro jednotlivé domény jsou vytvářené aplikací Hostmin. Menší problém nastává při generování statistik pro všechny domény najednou. Vytvoříme si proto skript `/root/scripts/genstats.sh`, který nám vygeneruje statistiku pro všechny domény s povoleným AWStats. Skript bude vypadat:

```
#!/bin/bash
DIR="/etc/awstats"
EXT="conf"

cd $DIR

for cfg in awstats.*.$EXT; do
if [ "$cfg" != "awstats.local.conf" ]; then
   site=`echo $cfg|sed -e "s/awstats.//g" | sed -e "s/.conf//g"`
   website=`echo $site|sed -e "s/www.//g"`

   /usr/lib/cgi-bin/awstats.pl -update -config="$site" -output \
-staticlinks > /var/www/"$website"/awstats/awstats.mysite.html
fi
done
```

Skript projde všechny konfigurační soubory AWStats pro jednotlivé domény uložené v adresáři `/etc/awstats` a spustí aktualizaci statistik. Upravíme oprávnění pro spuštění skriptu:

```
chmod 555 /root/scripts/genstats.sh
```

Posledním krokem je nastavení spuštěni skriptu každý den nebo tak často jak budeme chtít statistiku generovat. Nejlepší volbou bude spouštění skriptu plánovačem neboli cronem. Cron nastavíme na spouštění skriptu každý den o půlnoci. Spustíme proto příkaz[1]:

```
crontab -e
```

Tento příkaz otevřel konfigurační soubor cron aktuálního uživatele. Do něj zapíšeme:

```
0 0 * * * /root/scripts/genstat.sh &>/dev/null
```

Výše uvedený řádek znamená, že každý den o půlnoci se spustí skript v `/root/scripts/genstats.sh`.

To je z konfigurace AWStats vše. Aktuální verze během psaní této práce byla 6.7.

4 Aplikace Hostmin

V tuto chvíli je server připraven pro instalaci aplikace Hostmin. Předtím, než ji začneme instalovat, restartujeme server. Ujistíme se tak, že všechny služby, jako jsou Postfix, Apache, Courier, PowerDNS a MySQL si načtou nejnovější konfigurační soubory a mi jsme na nic nezapomněli. Aplikace je nahrána na CD, které je jako příloha k této práci.

Vývoj aplikace byl ve skriptovacím jazyce PHP a část v programovacím jazyce C. Aplikace je propojena s databázovým serverem MySQL. Jako podporu při programování byl použit PHP framework Kohana[5] ve verzi 2.3.4.

Na přiloženém CD jsou dva adresáře. Adresář s názvem `Hostmin`, který obsahuje zdrojové kódy celé aplikace. Dále pak druhý adresář s názvem `SQLhostmin`, který obsahuje všechny SQL skripty aplikace připravené pro import do MySQL databáze. Na CD se nachází tato kompletní práce ve formátu PDF s názvem souboru `dp-lis103-2011.pdf`.

4.1 Aplikace Hostmin

Prvním krokem instalace je import struktury databáze aplikace do databázového serveru MySQL. V adresáři `SQLhostmin` jsou dva soubory. První s názvem `structure-hostmin.sql` obsahuje strukturu databáze. Ten importujeme jako první. Druhý soubor `data-hostmin.sql` obsahuje základní data. Základními daty jsou výchozí nastavení aplikace, uživatelé a základní oprávnění. Import můžeme provést přes webové rozhraní aplikace PHPMyadmin, který si krátce popíšeme.

Ve webovém prohlížeči přejdeme k PHPMyAdminu na našem serveru například adresou `http://ipadresaserveru/phpmyadmin/`. Kde `ipadresaserveru` je konkrétní IP našeho serveru. Po přihlášení do PHPMyAdminu máme hned na úvodní stránce kromě informací o databázi volbu *Import*. Kliknutím na odkaz *Import* přejdeme na možnosti importu

[5] http://kohanaframework.org/ - vývoj verze 2.x byl ukončen, od verze 3.x je kompletně celý inovovaný

souboru. Nejdříve vybereme pro import soubor se s databází a strukturou `structure-hostmin.sql`. Jakmile bude vytvořena databáze, přejdeme do ní a importujeme soubor `data-hostmin.sql` se základními záznamy. Musíme ještě změnit heslo uživatele `hostmin`, který byl vytvořen s databází `hostmin`. V PHPMyAdminu pod záložkou oprávnění, změníme heslo pro uživatele s uživatelským jménem `hostmin`. Je důležité, aby měl uživatel `hostmin` oprávnění globální. Aplikace totiž bude vytvářet nové databáze a jejich uživatele.

Během konfigurace serveru jsme se zmiňovali o několika uživatelích pro balíky jako je Postfix, ProFTPd a PowerDNS. Ty jsme si vytvořili již v předešlých krocích. V tuto chvíli jim přidělíme konkrétní oprávnění. Pro uživatele `postfix`, nastavíme oprávnění `SELECT` pouze pro tabulky `mail_virtual_domains`, `mail_virtual_users`, `mail_virtual_aliases` v databázi `hostmin`. Pokračujeme dalším uživatelem, kterého jsme vytvořili při konfiguraci ProFTPd. Pro uživatele `proftpd` nastavíme oprávnění `SELECT`, `INSERT`, `UPDATE`, `DELETE` pro tabulky `ftpgroup`, `ftpquotalimits`, `ftpquotatallies`, `ftpuser` v databázi `hostmin`. Posledním uživatelem je `powerdns`. Pro tohoto uživatele nastavíme oprávnění na `SELECT`, `INSERT`, `UPDATE`, `DELETE` u tabulek `domains`, `records`, `supermasters` v databázi `hostmin`. Všechny výše uvedené oprávnění jsou vytvořeny z bezpečnostního hlediska tak, aby uživatelé s těmito oprávněními nemohli ovlivňovat jiné tabulky než ty, které potřebují ke správnému chodu vlastní aplikace.

Po nakonfigurování databáze můžeme přistoupit k instalaci a nastavení konfiguračních souborů aplikace Hostmin. Nejdříve se zaměříme na popis a rozdělení aplikace z hlediska jejich adresářů a souborů.

4.1.1 Instalace aplikace

V adresáři hostmin na přiloženém CD je celá aplikace rozdělená do několik adresářů. Kromě těchto adresářů jsou zde soubory `.htaccess` a `index.php`. Každý z těchto adresářů má svůj specifický význam:

- **application** – aplikace, základní rozhraní a šablony
- **hosts** –konfigurační soubory vytvořené aplikací Hostmin
- **media** – javascriptové soubory, obrázky a ikonky aplikace, soubory stylů
- **modules** – moduly aplikace
- **system** –knihovny a jádro frameworku Kohana
- **tmp** – adresář pro dočasné soubory

V adresáři `application` se nachází základní konfigurační soubory, šablony aplikace a některé pomocné soubory. Celá aplikace je rozdělena do několika modulů. V adresáři `modules` jsou do samostatných podadresářů rozděleny moduly aplikace Hostmin. Jednotlivé moduly popíšeme později.

První krokem instalace je zkopírování celého adresáře `hostmin` do adresáře na serveru, kde budou umístěny také veškeré webové stránky. Tento adresář je

nastavený v aplikaci Hostmin na `/var/www/`. Pro samotnou aplikaci si vytvoříme nového uživatele a změníme vlastníka včetně přístupových práv adresáře `/var/www/hostmin`:

```
useradd hostmin
chown -R hostmin:hostmin
chmod 755 -R /var/www/hostmin
chmod g+s /var/www/hostmin
```

Dále musíme upravit oprávnění pro jednotlivé adresáře uvnitř aplikace. Jedná se o adresáře pro dočasné soubory `/var/www/hostmin/hosts` a `/var/www/hostmin/tmp`:

```
chmod 777 /var/www/hosts
chmod 777 /var/www/tmp
```

Adresář pro soubory vyrovnávací paměti:

```
chmod 777 /var/www/application/cache
```

A jako poslední adresář pro soubory zaznamenávající případné chyby a varování aplikace:

```
chmod 777 /var/www/application/logs
```

Oprávnění musíme uzpůsobit také pro pár souborů. Konkrétně pro soubory `.htaccess` a `index.php`:

```
chmod 644 /var/www/hostmin/.htaccess
chmod 644 /var/www/hostmin/index.php
```

Předpokládá se, že aplikace poběží na subdoméně serveru, například administrace.example.com. Pokud by jste chtěli aplikaci provozovat v odděleném adresáři na serveru jako je například http://example.com/administrace/, musíme soubor .htaccess ještě upravit. Soubor používá Apache2 modul `rewrite`, který slouží mimo jiné také pro přepisování URL adres na tak zvané „hezké url“. To znamená, že URL adresa bude v lépe čitelném tvaru pro návštěvníky stránek a tím pádem snadněji zapamatovatelná[4]. V souboru je kromě jiných parametrů, také parametr `RewriteBase`, který přímo nastavuje odkud se bude URL adresa přepisovat. Základní nastavení je pouze / (lomítko) To platí pro situaci, kdy používáme subdoménu nebo pouze doménu, jako byl první příklad. Pokud bychom chtěli použít adresu jako v druhém příkladu (http://example.com/administrace/), musíme jako parametr `RewriteBase` dosadit hodnotu `/administrace/`. Nesmíme zapomenout na lomítko na začátku a na konci hodnoty parametru.

Poslední soubory, u kterých musíme nastavit práva jsou soubory v adresáři `/var/www/hostmin/modules/plugins/`. Soubory jsou dva a jedná se o malé aplikace, které se starají o zápis konfiguračních souboru, vytváření složek nebo případně vytváření uživatelů v systému. Obě tyto aplikace byly napsány

v jazyce C a zkompilovaný. Součástí adresáře jsou také zdrojové soubory, které mají stejný název a příponu .c. Jsou to soubory `apache_plg` a `mail_plg`. Oba jsou zkompilované pro použití přímo na serveru a v distribuci Debian. Pokud by jsme je potřebovali překompilovat, není nutné zadávat žádné speciální volby a můžeme použít příkaz:

```
gcc apache_plg.c –o apache_plg
gcc mail_plg.c –o mail_plg
```

K těmto souborům se ještě vrátíme později. V tuto chvílí jim jen změníme vlastníka, oprávnění a nastavíme SetUID[1]:

```
chown root:hostmin /var/www/hostmin/modules/plugins/apache_plg
chmod 775 /var/www/hostmin/modules/plugins/apache_plg
chmod u+s /var/www/hostmin/modules/plugins/apache_plg

chown root:hostmin /var/www/hostmin/modules/plugins/mail_plg
chmod 775 /var/www/hostmin/modules/plugins/mail_plg
chmod u+s /var/www/hostmin/modules/plugins/mail_plg
```

Veškerá oprávnění máme. Předtím než začneme měnit konfigurační soubory aplikace, musíme se rozhodnout na jaké URL adrese poběží. Musíme nejdříve vytvořit konfigurační soubor webového serveru Apache2 pro aplikaci Hostmin. Vytvoříme soubor s názvem `hostmin.vhost` v adresáři `/etc/apache2/sites-avaiable/` a vložíme do souboru následující řádky[1,4]:

```
<VirtualHost *:80>
      ServerName hostmin.example.com
      ServerAlias hostmin.example.com

      DirectoryIndex index.php
      DocumentRoot /var/www/hostmin

      CustomLog /var/www/hostmin/application/logs/ combined
</VirtualHost>

<Directory /var/www/hostmin>
      Options FollowSymLinks
      AllowOverride All
</Directory>
```

Značky `<VirtualHost *:80>` a `</VirtualHost>` definují virtuální rozhraní. Virtuální rozhraní dovolují Apache2 web serveru identifikovat požadavky nejenom na základě portu, ale také na základě IP. Díky protokolu HTTP 1.1 pak může Apache2 identifikovat spojení na základě jména domény, které je při požadavku na server obsaženo v hlavičce protokolu. Celé se to pak nazývá virtuální servery určené jmény. Tím odpadají omezení provozovat na jedné IP adrese jednu doménu. Díky identifikaci jmény můžeme na jednom stroji na jedné IP adrese, nebo více pokud je na serveru více síťových rozhraní s odlišnou IP adresou, provozovat desítky až stovky virtuálních serverů. V parametru

<VirtualHost *:80> říkáme Apache2 web serveru, že má naslouchat na všech síťových rozhraních a portu 80 pro doménu hostmin.example.com, která je uvedena v parametru ServerName. Hvězdička může být nahrazena za konkrétní IP adresu síťového rozhraní serveru. Parametrem ServerAlias může nastavit alternativu pro ServerName. DirectoryIndex pak řekne Apache2 serveru, který soubor se považuje za výchozí. Nejdůležitějším Parametrem je DocumentRoot, kterým definujeme adresář s obsahem pro konkrétní virtuální server a který bude také zobrazovat. Posledním parametrem je CustomLog. Apache2 web server zapisuje všechny přístupy k serveru do souboru uvedeného v CustomLog. U toho parametru je ještě nutné uvést druhou hodnotu, v našem případě combined, která určuje v jakém tvaru se mají přístupy návštěvníku stránek do souboru zapisovat.

Mezi <Directory> a </Directory> se zapisují parametry které ovlivní pouze zmíněný adresář /var/www/hostmin, ale také podadresáře a soubory. Adresář se zapisuje hned za otvírací značku stejně jak je uvedeno v příkladu. Parametr Options nastavuje jaké vlastnosti jsou k dispozici v daném adresáři. Hodnota parametru FollwoSymLinks říká Apache2 web serveru, že má následovat symbolické odkazy v daném adresáři. Poslední parametr, který je AllowOverride All sděluje, že pokud je v adresáři soubor .htaccess, dovoluje mu některé parametry přepsat a aplikovat. Více informací o jednotlivých parametrech najdete v oficiální dokumentaci webového serveru Apache2[6]. Pokud bychom chtěli používat zabezpečený protokol HTTPs, stačí do souboru vložit ještě jednou oddíl <VirtualHost *:80> se stejným obsahem a místo portu 80 uvést port 443. Dále do tohoto oddílu přidat informaci o námi vytvořeném SSL certifikátu, který jsme vytvářeli na začátku konfigurace apache2:

```
SSLEngine On
SSLCertificateFile /etc/apache2/ssl/hostmin.pem
```

Soubor máme vytvořený. Pro znovunačtení konfiguračních souborů včetně toho, který jsme právě vytvořili spustíme příkaz[1,4]:

```
a2ensite hostmin.vhost
/etc/init.d/apache2 reload
```

Prvním příkazem povolíme vytvořenou stránku (virtuálního hosta), která je v daném konfiguračním souboru. Jednou z vlastností Apache2 web serveru je, že umí podle souborů rozdělovat virtuální rozhraní na aktivní a neaktivní. Dokud není aktivovaný příkazem a2ensite, zůstává konfigurační soubor v adresáři /etc/apache2/sites-available/. Po aktivaci se v adresáři /etc/apache2/sites-enabled vytvoří symlink na konfigurační soubor do adresáře /etc/apache2/sites-available/. Druhý příkaz pak řekne Apache2 serveru, že si má načíst nové konfigurační soubory. Apache2 to udělá bez toho aniž by se musel restartovat nebo přerušit svou dosavadní práci. Protože aplikace Hostmin bude takovéto soubory při přidávání domén vytvářet sama,

[6] Oficiální dokumentace Apache2 web serveru na http://httpd.apache.org/docs/2.0/

musíme si zvolit jak bude probíhat jejich načítání Apachem. Nejlepším způsobem bude spouštění příkazu pro znovunačtení konfiguračních souborů periodicky pomocí cronu. Spustíme tedy příkaz[1]:

```
crontab -e
```

Otevře se cron aktuálního uživatele. V něm by již měl být zápis pro generování statistik AWStats a zasílání reportu o využití disku z předchozí konfigurace systému. Přidáme nový řádek:

```
0 */2 * * * /etc/init.d/apache2 reload &> /dev/null
```

Celý tento řádek znamená, že se bude každé dvě hodiny v celou spouštět příkaz `/etc/init.d/apache2 reload`, který načte znovu své konfigurační soubory.

Do cronu si ještě přidáme spuštění skriptu, který hlídá velikosti databáze. To nám zajistí php soubor, ověřující velikost databází, v adresáři `/var/www/hostmin/modules/plugins/dbquota.php`. Spouštění skriptu nastavíme také na každé dvě hodiny:

```
0 */2 * * * /usr/bin/php -f \
/var/www/hostmin/modules/plugins/dbquota.php &> /dev/null
```

Ve skriptu pro kontrolu velikosti databáze je ještě nutné nastavit heslo pro přístup do databáze na řádku č. 9 v nešifrovaném tvaru:

```
$mysql_pass  = "HESLO";
```

A nakonec nastavit přístupová práva k souboru:

```
chmod 755 /var/www/hostmin/modules/plugins/dbquota.php
```

Poslední krokem je upravení konfiguračních souborů aplikace. Nejdříve si pro editaci otevřeme soubor `/var/www/hostmin/application/config/config.php` a změníme řádek:

```
$config['site_domain'] = 'hostmin.example.com/';
```

Jedná se URL adresu přes kterou se bude přistupovat k naší aplikaci. Hodnota je stejná jako je uvedena v konfiguračním souboru Apache2 v parametru `ServerName`. Uvádíme ji bez protokolu. Ostatní parametry ponecháme stejné. Druhým konfiguračním souborem, který si otevřeme pro editaci je soubor `/var/www/hostmin/application/config/database.php`. V tomto souboru se nastavují hodnoty pro připojení k databázi:

```
$config['default'] = array
(
      'benchmark'      => TRUE,
      'persistent'     => FALSE,
      'connection'     => array
      (
```

```
            'type'     => 'mysql',
            'user'     => 'hostmin',
            'pass'     => 'HESLO',
            'host'     => 'localhost',
            'port'     => FALSE,
            'socket'   => FALSE,
            'database' => 'hostmin'
        ),
        'character_set' => 'utf8',
        'table_prefix'  => '',
        'object'        => TRUE,
        'cache'         => FALSE,
        'escape'        => TRUE
);
```

Hodnoty ponecháme ve výchozím stavu. Všechny by měli odpovídat našemu nastavení. Změníme jen hodnotu parametru `pass`, kde nastavíme správné heslo pro připojení k databázi uživatelem `hostmin`. Heslo se zadává v nešifrovaném tvaru.

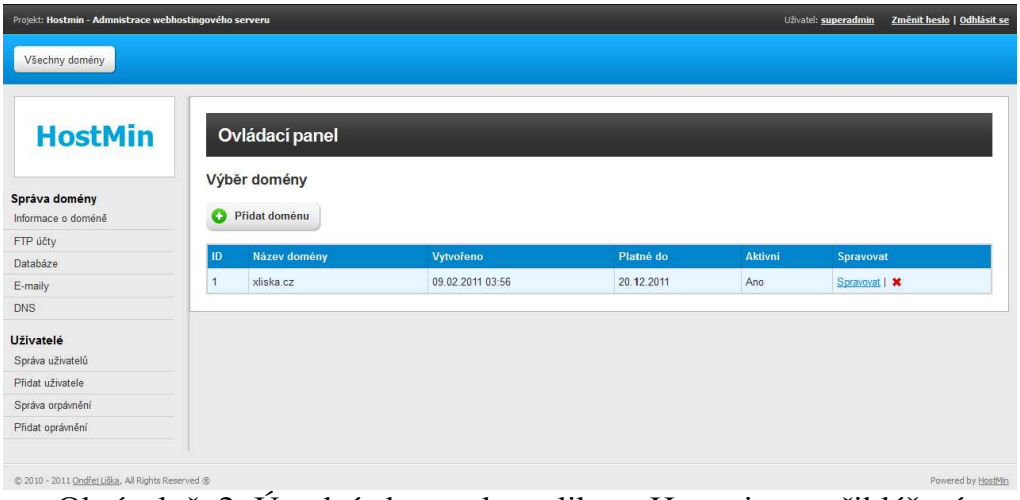
Aplikace je nakonfigurovaná. Pokud proběhlo vše v pořádku, můžeme k ní přistoupit přes webové rozhraní na adrese, kterou jsme si nastavili v konfiguračním souboru. Jako příklad jsme si uvedli adresu `http://hostmin.example.com`.

4.2 Obsluha aplikace

Server máme nastavený a aplikaci nainstalovanou. V několika malých kapitolách si teď popíšeme prostředí, chování a ovládání aplikace Hostmin. Aplikace je rozdělena do několika modulů a podle nich si popíšeme celou aplikaci. Aplikace může sloužit pro klienty hostingové služby nebo pouze pro správce serveru, který si tak významně zjednoduší práci při správě domén, databází, ftp účtu, emailových účtů nebo správě DNS záznamů. Při importu dat do databáze byl vytvořen pro přihlášení výchozí uživatel:

Uživatelské jméno: superadmin
Heslo: superadmin

Aplikace je graficky rozdělena na dvě hlavní části viz. obr. 2. V levé části se nachází vertikální navigace a v pravé části je hlavní okno aplikace.



Obrázek č. 2: Úvodní obrazovka aplikace Hostmin po přihlášení

4.2.1 Uživatelé

Součástí každé aplikace by měla být alespoň jednoduchá správa uživatelů. U této aplikace je jejich správa podstatně vylepšená. Jak už bylo zmíněno, celá aplikace je rozdělena do několika modulů. Systém, který je rozdělený do modulů, lze lépe spravovat s ohledem na oprávnění uživatelů. Do aplikace mají přístup pouze uživatelé, kteří jsou vytvoření aplikací Hostmin. Nejedná se o uživatele systému, ale pouze o uživatele aplikace. Každý uživatel má přidělenou právě jednu skupinu přístupových práv, dále jen oprávnění. Oprávnění lze spravovat z aplikace Hostmin v navigaci pod nabídkou *Správa oprávnění*.

Po kliknutí na nabídku *Správa oprávnění* se zobrazí výpis s aktuálně vytvořenými oprávněními. Základní oprávnění jsou `sroot` a `root`. Tyto dvě nelze odstranit a upravit jde pouze oprávnění `root`. Výchozí uživatel `superadmin` má přiřazeno oprávnění `sroot`, ktere disponuje neomezeným přístupem ke všem modulům. Jednotlivá oprávnění jsou rozdělená do modulů aplikace. Pokud nějaký modul v něm není obsažený, nemůže uživatel s tímto oprávněním vykonávat žádnou akci v daném modulu uvedené v tab. 3.

Název modulu	Popis modulu
databases	správa databází MySQL v závislosti na doméně
dns	správa DNS záznamů
domains	správa VirtualHost Apache2 web serveru
ftp	správa FTP účtů v závislosti na doméně
mails	správa emailových účtů v závislosti na doméně
users	správa uživatelů aplikace Hostmin a jejich oprávnění

Tabulka č. 3: Moduly aplikace Hostmin

Kromě oprávnění `sroot`, lze u všech dalších oprávnění měnit a upravovat moduly a jejich akce viz. tab. 4. Jejich hodnota je *Povoleno* nebo *Zakázáno*.

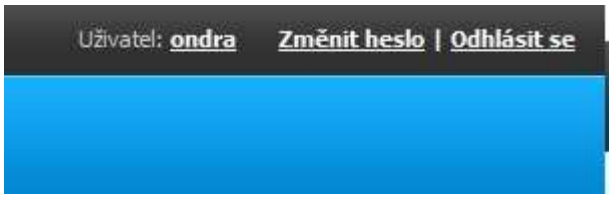
Název akce	Popis
Vytváření	Vytvořit záznam (např. přidání domény, vytvoření FTP účtu)
Editace	Upravit záznam (např. změna FTP účtu, změna databáze)
Mazání	Odstranit záznam (např. odstranění domény)
Nastavení	Změna nastavení modulu (pouze u DNS, výchozí nastavení)

Tabulka č. 4 Omezení oprávnění na akce

Nové oprávnění můžeme vytvořit kliknutím na nabídku navigace *Přidat oprávnění*. Po zobrazení formuláře vyplníme název oprávnění a jeho popis. V seznamu modulů si vybereme takové, které chceme přidat, zatržením políčka *Přidat*. Dále pak vybereme akce, které chceme u každého modulu povolit zatržením příslušného políčka. Uložením formuláře vytvoříme nové oprávnění. Oprávnění smí vytvářet pouze uživatelé s oprávněním `sroot` a `root`.

Vytváření a spravování oprávnění je jedna ze dvou částí práce s uživateli. Druhá část je vytváření uživatelů a jejich správa. V navigace v části *Uživatelé* je také k dispozici nabídka *Správa uživatelů* a *Přidat uživatele*. Pod nabídkou *Správa uživatelů* najdeme výpis všech uživatelů aplikace Hostmin. Uživatelé s oprávněním `sroot` nelze odstranit. Odstranit uživatele může pouze ten, který má k tomu dostatečná oprávnění a zároveň může odstranit pouze ty, kteří mají nižší úroveň oprávnění než má daný uživatel. Stejné pravidlo platí pro editaci těchto uživatelů a také pro vytváření. Úroveň oprávnění je dána číslováním v tabulce *Správa oprávnění*, kde nižší číslo znamená vyšší úroveň.

Vytvoření nového uživatele provedeme kliknutím na položku *Přidat uživatele*. Při vytváření nového uživatele registrujeme základní údaje jako je *Uživatelské jméno*, *Heslo* a *Role* (oprávnění). Ostatní údaje jsou nepovinné. Každý uživatel si také může měnit své osobní údaje a heslo. Pro změnu hesla a osobních údajů, je tato možnost vyhrazena v pravém horním rohu aplikace jako na obr. 3. Pro změnu profilu a hesla stačí kliknout na uživatelské jméno, pro změnu pouze hesla, na nabídku změnit heslo.



Obrázek č. 3: Osobní navigace přihlášeného uživatele

Z hlediska uživatelů a jejich oprávnění jsme probrali vše a můžeme pokračovat popisem ostatních modulů.

4.2.2 Domény

Celý modul domény slouží pro přidávání domén webového serveru Apache2. Výpis domén se po přihlášení zobrazuje na úvodní stránce aplikace. Zobrazují se pouze domény, které jsou přiřazeny k danému uživateli nebo všechny, pokud má uživatel oprávnění vytvářet nové záznamy. V případě, že disponuje oprávněním přidávat záznamy, máme k dispozici tlačítko pro přidání nové

domény. Jestliže se nacházíme v jiné části aplikace, dostaneme se na výpis domén kliknutím na volbu *Všechny domény*, kterou najdeme vlevo nahoře nad logem aplikace Hostmin. Kromě modulu Uživatelů a DNS jsou na výběru domény závislé všechny ostatní moduly. FTP účty, MySQL databáze a emailové účty se vždy vytvářejí ke konkrétní doméně.

Přidat novou doménu lze kliknutím na volbu *Přidat doménu*. Nejdříve vyplníme zobrazený formulář, kde je povinou položkou pouze název domény (*Doména*). Doménu můžeme také přiřadit konkrétnímu uživateli, uděláme to výběrem v poli *Klient*. Pokud má některý uživatel přiřazenou doménu, může ji kompletně spravovat i se všemi souvisejícími moduly bez ohledu na to, zda má právě tyto oprávnění. Položka ve formuláři *Limit* stanovuje omezení diskového prostoru pro doménu. Položka *Platné do* má pouze informativní charakter. Další položky *.htaccess*, a*wstats* a *Aktivní*, mají vliv na vytvoření konfiguračních souboru.

Při vytvoření nové domény dochází k vytvoření konfiguračního souboru webového serveru Apache2 v adresáři `/etc/apache2/sites-available/`. Konfigurační soubor obsahuje standardní informace o virtuálním serveru pro nově vytvořenou doménu. Jestliže je zatržena také volba *.htaccess*, je do konfiguračního souboru přidána informace o povolení .htaccess souborů v doméně. Pokud je zatržena volba *awstats*, dojde také k vytvoření konfiguračního souboru pro stejnojmenný software v adresáři `/etc/awstats/`, který jsme si popsali už dříve. Konfigurační soubor pro AWStats má název `awstats.example.com.conf`. U každé domény také registrujeme dva stavy. Pokud je při vytvoření zatrženo pole *Aktivní*, aktivuje se konfigurační soubor webového serveru Apache2 příkazem `a2ensite domena`, kde parametr `domena` je název domény a zároveň konfiguračního souboru, který byl vytvořeny v adresáři `/etc/apache2/sites-available`. Pokud tato volba není zatržena k jeho aktivaci nedojde. V případě že editujeme doménu, která je aktivní a volbu *Aktivní* odškrtneme, doména bude deaktivována příkazem `a2dissite domena`.

V konfiguračním soboru virtuálního serveru je uveden parametr `DocumentRoot`. Jeho význam jsme si vysvětlili při instalaci aplikace Hostmin. Proto při vytvoření nové domény musí dojít logicky k vytvoření adresáře, který je uveden v tomto parametru. Adresáře domén jsou vytvářeny v základním adresáři `/var/www/`. Pravidla jsou nastavena tak, že adresář domény má stejný název jako doména, takže například pro doménu *example.com* bude vytvořeny adresáře `/var/www/example.com/`. V tomto adresáři se ještě vytvoří výchozí adresář `www` pro webové stránky, `awstats` pro vygenerované statistiky a adresář `logs`, který obsahuje soubor s hlášením přístupu k doméně. Soubor s hlášením je generovaný webovým server Apache2 a používá se také pro generovaní statistiky aplikací AWStats. Parametr `DocumentRoot` bude mít tedy hodnotu `/var/www/example.com/www/`.

Abychom zabezpečili oprávnění pro adresář domény, je nutné změnit ještě vlastníka, skupinu a oprávnění. Během přidávání domény dojde k vytvoření systémového uživatele, který má jméno shodné s názvem domény. Uživatel bude vytvořený s domovským adresářem v adresáři webových stránek a login

shell nastavený na `/bin/false`. `Domena` je nahrazena konkrétním názvem domény, například `example.com`:

```
useradd -d /var/www/domena -s /bin/false domena
```

Po vytvoření systémového uživatele je provedena změna vlastník a skupiny adresáře domény na právě vytvořeného systémového uživatele.

Nejdůležitějším u vyvíjení tohoto modulu bylo vymyslet způsob, jak z aplikace Hostmin spouštět některé z nutných systémové příkazy. Většina z těchto příkazu potřebovala oprávnění systémového uživatele root. Apache2 web server spouští PHP skripty pod uživatelem jiným uživatele, v distribuci Debian je to uživatel www-data, a nemůže tak zapisovat do složky webového serveru. Z hlediska bezpečnosti, změnit uživatele www-data na systémového uživatele root by byl tento krok velmi nevhodný.

Zvoleným řešením nakonec bylo vytvoření souboru, který bude tyto příkazy spouštět a nastavení oprávnění SetUID na tento soubor. Takové oprávnění znamená, že soubor bude spuštěný s UID vlastníka souboru. Pokud tedy nastavíme jako vlastníka systémového uživatele `root`, může soubor s tímto oprávněním spouštět příkazy jako by se jednalo o uživatele `root`. Vlastnost SetUID funguje pouze u binárních souborů. Vhodným kandidátem pro vytvoření binárního souboru byl programovací jazyk C. Zkompilováním souboru napsaného v jazyce C vznikne binární soubor, který můžeme pro náš účel použít.

Pro vytváření, editaci a mazání domén webového serveru Apache2, ale také pro vytváření a mazání konfiguračních souboru AWStats, se používá soubor s názvem `apache_plg` umístěný v `/var/www/hostmin/modules/plugins/`. Oprávnění SetUID i vlastníka jsme již nastavili při instalaci aplikace Hostmin.

Po vytvoření domény v aplikaci Hostmin, jsme automaticky přesměrováni zpět na výpis všech domén. Abychom mohli pokračovat, musíme si vybrat, kterou doménu chceme aktuálně spravovat. Kromě odkazu *spravovat* pro výběr domény, je zde ikonka pro odstranění domény ze serveru. Při odstranění dojde ke smazání konfiguračních souboru, databází, emailu, ftp účtu, ale také k odstranění adresáře pro webové stránky. Odstraněný bude i systémového uživatele, který má stejný název jako doména. Po kliknutí na volbu *spravovat* se zobrazí informace o doméně a výpis z registru domén, viz obr. 4, pokud je doména již zaregistrovaná.

Obrázek č. 4: Detail domény xliska.cz v aplikaci Hostmin

Stejné položky které jsme zadávali při vytváření můžeme změnit kliknutím na tlačítko Upravit informace o doméně. Změnit můžeme kterékoliv údaje jako při vytváření kromě údaje *Doména*.

V okamžiku kdy se nacházíme v části informace o doméně je v navigaci navíc k dispozici nabídka *Subdomény*. Po kliknutí na ní, přejdeme na výpis subdomén pro právě spravovanou doménu. Přidání subdomény k doméně provedeme kliknutím na *Přidat doménu*. Zobrazí se formulář pro vytvoření nové subdomény. Povinou položkou je pouze název subdomény *Subdoména*. Název se zadává bez doménového jména, které bude automaticky doplněno. Můžeme si také vybrat některý z druhů přesměrování na jinou adresu v poli *Přesměrovat*. Možnosti odpovídají možnostem v tab. 5.

Typ přesměrování	Popis
Bez přesměrování	Subdoména nebude přesměrována
Bez příznaků	Stránka bude tak zvaně podstrčená
L	Jedno přeložení adresy, první vyhovující podmínka
R	Dočasné přesměrování (302)
R, L	Kombinace dvou předchozích

Tabulka č. 5: Možnosti přesměrování subdomény v aplikaci Hostmin

Pokud si vybereme některý z druhů přesměrování, musíme také doplnit položku P*řesměrovat na* jako cílovou adresu přesměrování, kterou může být i jiná doména, než na které vytváříme subdoménu. Položka *Povolit .htaccess* a *Aktivní* mají stejný význam jako u vytváření domény. Po vytvoření se dostáváme zpět na výpis existujících subdomén. Zároveň dojde k vytvoření adresáře `/var/www/example.com/nazevsubdomeny/`, kde `nazevdomeny` je nahrazen skutečným názvem domény, a upravení konfiguračního souboru Apache2 web

serveru pro zvolenou doménu. Informace o subdoméně se musí také zaznamenat do tohoto konfiguračního souboru. Subdoménu můžeme také upravit kliknutím na ikonku tužky. Kromě názvu můžeme změnit, kteroukoliv položku. Samozřejmostí je také její odstranění. Při odstranění dojde k odstranění odpovídajícího adresáře subdomény včetně jeho obsahu.

Vytváření adresáře subdomény a úpravě konfiguračního souboru domény opět slouží soubor `apache_plg`, stejně jako u vytváření domény. Změny se projeví až po novém načtení konfiguračních souborů. Znovunačtení probíhá periodicky tak, jak jsme si nastavili během instalace aplikace Hostmin do cronu.

4.2.3 FTP účty

Seznam FTP účtu pro spravovanou doménu najdeme pod stejnojmennou položkou v navigaci *FTP účty*. Zvolením této položky se dostaneme na seznam všech vytvořených FTP účtu pro danou doménu. Seznam pro nově vytvořenou doménu je prázdný. Vytvoření nové FTP účtu provedeme kliknutím na tlačítko *Vytvořit FTP účet*. Zobrazí se nám formulář ve které musíme vyplnit několik položek. Povinné položky jsou *Uživatelské jméno* a *Heslo* pro přihlášení k FTP serveru. *Výchozí adresář* je adresářem ve kterém bude uživatel po přihlášení k FTP serveru. V tomto adresáři bude přihlášený uživatel uzamčen. To znamená, že nemá přístup do nadřazených adresářů. Pokud není cesta výchozího adresáře nijak určena, je automaticky výchozí adresář nastaven na „/“. Lomítko však v tomto případě znamená adresář webových stránek `/var/www/domena/`, kde `domena` je opět skutečný název domény. Lze tedy zadávat pouze cesty v rámci tohoto adresáře.

Diskový prostor pro jednotlivé účty není nijak limitován. Omezený je pouze prostor pro celou doménu. Toto omezení se nastavuje už při vytváření domény.

FTP účty můžeme také upravovat kliknutím na ikonku tužky. Upravit lze všechny položky, které jsme měli k dispozici při vytváření a to včetně *Uživatelského jména* a *Hesla*. Pokud budeme účet upravovat a necháme položku *Heslo* prázdnou, zůstane zachováno původní heslo. Pro odstranění FTP účtu slouží stejně jako v předchozích případech ikonka červeného křížku. Vytváření a mazání FTP účtu může pouze uživatel kterýma k tomu dostatečná oprávnění, nebo uživatel, který má ke svému účtu přiřazenou danou doménu.

U FTP účtu není zapotřebí provádět žádné systémové příkazy jako tomu bylo u domén. Používáme balík ProFTPd spolu s modulem pro práci s MySQL databázi. ProFTPd máme nastavené pro spolupráci přímo s databázi aplikace Hostmin. Veškeré účty jsou proto vedeny v této databázi a jakékoliv změny se projeví ihned bez nutnosti znovuzavedení ProFTPd.

4.2.4 Databáze MySQL

Již existující MySQL databáze najdeme pod nabídkou *Databáze*. Zvolením položky se nejprve dostaneme na seznam vytvořených databázi. Stejně jako

u FTP účtu se jedná o databáze vytvořené pro spravovanou doménu. Při vytvoření domény nedochází k vytvoření MySQL databáze a proto je seznam prázdný dokud nějakou databázi nevytvoříme. Tlačítko pro vytvoření se opět nachází nad výpisem databází. Po kliknutí na *Vytvořit databázi* se nám zobrazí formulář pro vytvoření nové databáze.

Povinné položky ve formuláři jsou *Název DB*, *Uživatel* a *Heslo*. Aby nedocházelo ke shodě názvů databázových jmen v rámci všech domén, název databáze začíná vždy prefixem *webIDdb*. ID v prefixu je nahrazeno jedinečným identifikátorem pro doménu. Stejná situace by mohla nastat u uživatelů databáze. Aby nedošlo ke shodě uživatelských jmen, každé uživatelské jméno tak začíná prefixem *webIDu*. Opět je ID v prefixu uživatelské jména jedinečný identifikátor domény. V obou případech je tento identifikátor číslo. Při konfiguraci MySQL serveru jsme povolili vzdálený přistup v konfiguračním souboru databáze. K tomu aby jsme mohli tuto možnost využívat musíme také povolit vzdálené přihlášení uživatele. Během vytváření databáze je k tomu určena položka *Vzdálený přistup*. Pokud ji necháme nezatrženou, můžeme se k databázi přihlásit například přes PHPMyAdmin nebo z PHP skriptu umístěného na stejném serveru jako MySQL databázový server. V případě, že volbu Vzdálený přístup zatrhneme, můžeme s k databázi přihlásit i vzdáleně například ze skriptu, který je umístěný na jiném serveru. Poslední položka formuláře je položka *Aktivní*. Pokud položku nezatrhneme, databáze se vytvoří, ale nemůžeme se k ní přihlásit.

Poslední položkou je pole omezení velikosti databáze s názvem *Limit*. Výchozí limit je 100 MB a pole smí editovat pouze uživatelé s dostatečným oprávněním.

Po vytvoření se opět dostaneme na výpis již existujících databází. Opět můžeme využít upravení databáze. Změnit můžeme položky *Uživatel*, *Heslo*, *Vzdálený přístup* a *Aktivní*. Název databáze je již neměnný. Databázi lze také odstranit. Před odstraněním se aplikace dotáže uživatele, zda chce opravdu databázi odstranit. Po potvrzení, je databáze nenávratně odstraněna.

Velikost databáze je hlídána skriptem, u kterého jsme nastavili spouštění do cronu během instalace aplikace Hostmin. Je to jednoduchý php skript, který kontroluje velikost databáze s jejím limitem a v případě překročení velikosti databáze odebere uživateli přiřazenému k této databázi možnost vytvářet a vkládat záznamy (`CREATE, INSERT`). Jakmile bude databáze zmenšená pod velikost jejího nastaveného limitu bude nejpozději během dvou hodin možnost vkládat a vytvářet záznamy obnovena. To záleží na periodě spouštění skriptu.

Všechny akce s databázemi jsou prováděny hned a pokud nedojde k nějaké neočekávané chybě, projevují se do několika vteřin. Výjimkou je změna omezení velikosti databáze, kdy probíhá její kontrola každé dvě hodiny respektive podle periody v cronu.

4.2.5 Emaily

Nedílnou součástí hostingové služby jsou doménové emaily. Vytvoření emailového účtu nebo jen emailového aliasu můžeme kliknutím na položku

Emaily v navigaci aplikace Hostmin. Jako u všech předchozích modulů se zobrazí výpis již existujících emailových adres, který je pro nově vytvořenou doménu prázdný. Emailové účty můžeme vytvářet pouze pro aktuálně spravovanou doménu.

Založení emailu provedeme kliknutím na tlačítko *Vytvořit emailový účet*. Ihned se nám zobrazí formulář pro vytvoření nového emailu. Povinné položky jsou *Uživatel*, *Email* a *Heslo*. Uživatelské jméno slouží jen pro informaci, jako odlišení jednotlivých účtů například jménem. Není nikde použitu pro přihlašování. Jako údaj pro přihlášení slouží celá emailová adresa. Do položky *Email* se zadává pouze název který je před zavináčem. Zbytek bude automaticky doplněn podle aktuálně spravované domény. Položka *Kvóta* je přístupná jen pro uživatele s oprávněním vytvářet emailové účty. Uživatelé s přiřazenou doménou a bez oprávnění vytvářet emailové účty mají automaticky přidělenou kvótu 100 MB. Poslední položkou ve formuláři je položka *Aktivní*. Položka nemá vliv na vytvoření emailového účtu, ale pouze na přihlášení k tomuto účtu. Pokud není účet aktivní, nelze se k němu přihlásit.

Formulář úpravy emailového účtu disponuje stejnými položkami jako u vytvoření. Výjimkou je pole *Email*, které již nelze změnit. Pro zbylá pole platí stejné pravidla jako při vytváření.

Ve chvíli kdy se nacházíme v části aplikace pro vytváření emailů, máme možnost v navigaci kliknout na položku *Emailový alias*. Alias je schránka, která je nasměrovaná na jinou schránku. To znamená, že pokud si vytvoříme alias například pepa@example.com a emailový učet josef@example.com, jsou emaily zaslané na adresu pepa@example.com doručeny na adresu josef@example.com. V aplikaci Hostmin si můžeme alias vytvořit kliknutím na *Vytvořit emailový alias*.

Formulář nabízí pouze tři položky. První položka je název *Alias*. Alias tvoří jen část před zavináčem. Zavináč a doména budou automaticky doplněny. Speciální variantou aliasu je možnost zapsat znak hvězdičku do pole *Alias*. Hvězdička má jako symbol v tuto chvíli význam pro cokoliv. To znamená, že všechny emaily zaslané na jakýkoliv email v aktuálně spravované doméně (example.com), budou nasměrovány na adresu uvedenou v *Přeposlat na*. Jedná se o tak zvaný doménový koš. Pokud není takovýto koš vytvořen, emaily poslané na neexistující emailový učet, ale existující doménu, jsou automaticky zahozeny. Vytvoření doménového koše se moc nedoporučuje, protože se poté hromadí vetší množství nevyžádané pošty. V poli *Přeposlat na* můžete zadat více než jednu emailovou adresu a nemusí být v rámci spravované domény. Pokud jich uvádíte více, musí být oddělené čárkou. Na emailové adresy uvedené v tomto poli, bude pošta z uvedeného aliasu nasměrována. Poslední položka s názvem *Aktivní* určuje stav nasměrování. Pro doručování pošty na vybrané emailové adresy musí být alias aktivní. Opět je zde i možnost upravit vytvořený alias. Upravit můžeme všechny položky, které jsou nabízeny při vytvoření aliasu. Samozřejmostí je také možnost alias odstranit.

Pro připojení můžeme použít službu POP3, IMAP nebo jejich šifrované SSL verze. Jako hostitel je původní název serveru. Případně si pomocí DNS můžeme nasměrovat například vlastní doménu na jméno serveru.

Jako emailový server jsem si nainstalovali Postfix s Courierem Mail server. Oba balíky jsme nakonfigurovali pro spolupráci s MySQL databází a proto se změny v emailových účtech nebo aliasech projevují ihned. Jediný problém, který bylo nutné vyřešit je vytvoření emailových schránek přímo na disku serveru, do které se zapisují doručené, ale také odeslané emaily. K tomu nám poslouží soubor `mail_plg`, který je umístěný ve `/var/www/hostmin/modules/plugins/`. Je to obdobný postup jako u vytváření domén. Soubor `mail_plg`, jak již bylo zmíněno při instalaci aplikace Hostmin, je malá aplikace, která byla vytvořená zkompilováním zdrojového souboru napsaném v jazyce C. Zároveň je nastaveno na aplikaci SetUID a vlastníkem je uživatel `root`. Emailové schránky jsou souborem `mail_plg` vytvářeny v adresáři `/home/vmail/`. Pro každou doménu je zde adresář, který má název stejný jako doména. V adresáři domény jsou pak další adresáře podle jednotlivých emailových schránek. Celá schránka například pro emailovou adresu pepa@example.com je umístěna v adresáři `/home/vmail/example.com/pepa/`.

4.2.6 DNS

Posledním modulem aplikace Hostmin, který nám zbývá popsat je správa DNS záznamu. Je to jediný modul, který není závislí na aktuálně spravované doméně. Spravovat záznamy mohou uživatelé s dostatečným oprávněním pro práci s modulem DNS nebo ti uživatelé, kteří mají přiřazený některou ze zón. Na seznam zón v DNS se dostaneme zvolením položky DNS v navigaci aplikace. Jako u všech ostatních modulů můžeme vidět nejdříve výpis, tentokrát vedených zón v našem DNS serveru. Novou zónu vytvoříme tlačítkem *Přidat zónu*. Zobrazený formulář obsahuje pouze čtyři pole. *Název zóny* a *typ* jsou povinné položky. *Název zóny* zadáváme jako doménu druhého řádu. *Typ* viz. tab. 6, určuje jak bude zóna dále replikovaná nebo ověřovaná.

Typ	Popis
NATIVE	Výchozí nastavení – o replikaci se stará databáze
SLAVE	Záznam je pravidelně ověřován u master DNS serveru
MASTER	Změna záznamu je automaticky oznámena slave DNS serverům

Tabulka č. 6: Typy replikace domén PowerDNS serveru

Pokud vybereme jako *Typ* položku SLAVE, musíme také vyplnit pole *Master* IP adresou master DNS serveru, u kterého se bude záznam ověřovat, případně aktualizovat. Poslední položka *Klient* přiřazuje doménu ke konkrétnímu uživateli.

Po vytvoření jsme automaticky přesměrovaní zpět na seznam zón. Pokud jsme zvolili jako typ MASTER nebo NATIVE, můžeme se podívat na detail zóny. Ta obsahuje již automaticky vytvořený SOA záznam. Sériové číslo tohoto záznamu

je vytvořeno jako složení číslic rok, měsíc, den a id domény. SOA záznam již obsahuje veškeré potřebné informace a nelze ho upravit. Pokud bychom chtěli tyto informace změnit, musíme upravit nastavení DNS modulu a zónu znovu vytvořit. Do nastavení se dostaneme zvolením položky *Nastavení*, která je k dispozici v okamžiku kdy se nacházíme v části DNS. Nastavení může upravovat pouze osoba, která má k tomu dostatečná oprávnění. Klienti, kteří mají přiřazenou nějakou doménu tato nastavení měnit nemohou.

Zónu můžeme také upravit nebo odstranit a to analogicky jako u předchozích modulů. Položku, kterou nelze změnit nebo upravit je *Název zóny*. Ostatní, které jsou stejné jako při vytváření zóny lze libovolně měnit.

Pokud se nacházíme v detailu zóny máme možnost přidat do zóny zdrojové záznamy. Záznam přidáme kliknutím na tlačítko *Přidat záznam*. Ihned se nám zobrazí formulář pro přidání záznamu. V jednoduchém formuláři vyplníme pole *Záznam*, vybereme *Typ* záznamu a doplníme pole *Hodnota*. Na výběr jsou nejběžnější typy záznamů A, AAAA, C-NAME, MX, NS a PTR. Pro PTR záznamy musí být vytvořena nová zóna (např. 3.2.1.in-addr.arpa) a v ní vytvoříme PTR záznam (např. *4*, který odpovídá IP 1.2.3.4 která směřuje např. na www.example.com). Pole *TTL* je zkratka pro *Time To Live* a určuje platnost záznamu po kterou je záznam uchovávaný v cache paměti DNS serveru. Pole priorita je platné jen pro záznamy typu MX, kterých může být více a můžeme určit jejich prioritu. Priorita je přirozené číslo a menší číslo znamená větší prioritu. Obvykle se používají čísla 10 a 20. Po vytvoření jsme opět automaticky přesměrováni na výpis záznamu v dané zóně, pro kterou jsem záznam vytvářeli. Záznamy můžeme také upravovat a odstranit. U záznamu můžeme změnit všechny jeho položky kromě názvu záznamu. Všechny zóny i záznamy jsou vedeny v MySQL databázi a změny v záznamech a zónách se na serveru projevují ihned. Nějakou dobu však trvá než se změna propaguje do dalších DNS serverů.

5 O vývoj aplikace Hostmin

Jak jsem se již zmiňoval v předešlých kapitolách, aplikace je postavena na skriptovacím jazyku PHP a databázi MySQL. Současně jsem pro vývoj použil frameworku Kohana. S tímto frameworkem jsem měl možnost už několikrát pracovat. Značně urychluje vývoj webových aplikací. Především vytváření formulářů, jejich validace a jiných rutinních operací jako je práce s databází. Aplikace se píše velmi snadno a rychle. Kohana je také postavena na architektuře MVC, kdy je aplikační logika oddělená od datového modelu a uživatelského rozhraní. Ve všech směrech jsem tento způsob vývoje také dodržoval a proto i aplikace Hostmin splňuje architekturu MVC. Framework Kohana nevynucuje dodržování této architektury, alespoň ve verzi 2.x.

Databáze aplikace Hostmin obsahuje tabulky, které jsou používány pro některé balíky již standardně. Je tomu tak například u tabulek pro ProFTPd nebo u tabulek pro PowerDNS. Ty jsou maximálně rozšířeny o jeden nebo dva atributy, jinak odpovídají obvykle uváděné struktuře pro dané balíky. Ostatní tabulky jsou buď částečně nebo výrazně upravené. Například u modulů domény, subdomény, oprávnění uživatelů nebo modulu pro správu databází.

Struktura tabulek je převážně jednoduchá a názvy jednotlivých sloupců jsou jasné a výstižné.

Během vývoje aplikace bylo důležité poměrně podrobně nastudovat jak fungují jednotlivé balíky a především jejich moduly pracující s MySQL databázi. U některých balíku, které nepodporovali práci s MySQL databází bylo nutné vymyslet jiné řešení jak spouštět systémové příkazy pro jejich konfiguraci. Možností jak toho dosáhnout je několik. Například spouštění určitých skriptů PHP nebo jiných pod uživatelem `root`, například pomocí cronu, které by kontrolovali změny v databázi. Pokud by byla nějaká změna, přímo ze stejného skriptu by byly zpracovány, právě spouštěním nějakých systémových příkazů. Tato varianta se mi však nezdála příliš bezpečná nebo výhodná. Především častou kontrolou databáze by se samotná databáze zbytečně zatěžovala. Jiný způsob by mohl být založený na interaktivním spouštění systémových příkazů. Vždy by se zadal požadavek na spuštění příkazu například přes sudo a zároveň by se automaticky uvedlo heslo uživatele `root`. Tato varianta má nevýhodu ukládání hesla uživatele `root` do skriptu. Variantou k sudo by mohlo být definování jednotlivých příkazu do soubor `/etc/sudoers`, ve kterém se uvádí kdo a co může spouštět s právy `root` nebo případně jinými právy. Nakonec jsem jako nejvhodnější variantu vybral použití speciálních přístupových práv SetUID u souboru. Nevýhoda však byla, že SetUID lze nastavit pouze na spustitelný binární soubor a aplikace, která má nastaveno SetUID musí být velmi bezpečná. Jinak by mohlo dojít k jejímu zneužití.

Takové dvě malé aplikace jsem vytvořil v programovacím jazyku C a následně zkompiloval do spustitelného binárního souboru. Tomu jsem nastavil jako vlastníka uživatele `root` a speciální přístupová práva SetUID. O obou těchto aplikacích jsem se již zmiňoval. Aby byli obě dostatečné bezpečné, systémové příkazy jsou velmi omezené a především závisle na volbách při spuštění. Například pro vytvoření nové domény jako virtuální server se spouští z aplikace Hostmin aplikace `apache_plg` umístěná v adresáři `/var/www/hostmin/modules/plugins/` následujícím způsobem:

```
/var/www/hostmin/modules/plugins/apache_plg create example.com 1
```

Volba `create` znamená, že se má vytvořit nový konfigurační soubor pro doménu `example.com`. Jednička na konci znamená, že nový virtuální server `example.com` má byt také hned aktivován. Aplikace předpokládá již předem vytvořený konfigurační soubor v adresáři `/var/www/hostmin/hosts/example.com`. Tento soubor je vytvořený aplikací Hostmin. Následně provede kroky jako je zkopírování souboru do adresáře `/etc/apache2/sites-available/`, vytvoření nového uživatele, vytvoření adresáře webových stránek, odstranění konfiguračního souboru vytvořeného aplikací Hostmin a nakonec aktivaci virtuálního serveru `example.com`. Jednička na konci pak může být nahrazena nulou. To by znamenalo, že virtuální server nebude aktivovány nebo případně bude deaktivovaný v případě, že předtím existoval a je prováděná aktualizace virtuálního serveru. Další akceptované volby jsou:

```
apache_plg update example.com 1
apache_plg delete example.com
apache_plg createsubdomain example.com subdomain
apache_plg updatesubdomain example.com subdomain
apache_plg deletesubdomain example.com subdomain
apache_plg awstats example.com 1
apache_plg quota example.com 10000 20000
```

Samozřejmě jsou volby `example.com`, `subdomain`, nebo `10000`, `20000` nahrazeny konkrétními hodnotami. Celá aplikace dokáže vytvářet konfigurační soubory pro AWStats, nastavovat diskové omezení nebo vytvářet, aktualizovat a mazat subdomény a domény. U diskového omezení jsou pak hodnoty `10000` a `20000`, hodnotami pro takzvaný lehký limit, který může být dočasně překročený a pro takzvaný tvrdý limit, který být překročený nemůže.

Na obdobném principu funguje i druhý spustitelný binární soubor s nastaveným SetUID s názvem `mail_plg`. Ten je určený především pro vytváření emailových schránek. Akceptuje však pouze dvě možnosti a to vytvoření:

```
mail_plg create example.com info
```

a nebo:

```
mail_plg delete example.com info
```

První volba vytvoří emailovou schránku `info` na disku pro doménu `example.com`. Jinak řečeno vytvoří adresář, základní soubory a strukturu pro zprávy typu `maildir` v adresáři `/home/vmail/`. Konkrétně adresář `/home/vmnail/example.com/info/`. Druhým příkazem s volbou `delete` tuto schránku odstraní. Odstranění se provádí bez jakékoliv zálohy, tedy nenávratně. Samozřejmostí je nahrazení voleb `example.com` a `info` za konkrétní doménu.

6 Závěr

V této práci jsem představil a popsal konfiguraci několika softwarových balíku pro distribuci operačního systému Debian GNU/Linux ve verzi 5.0 (Lenny), která slouží ke zprovoznění hostingové serveru. Především jsem se zaměřil na takový způsob instalace a takový software, který spolupracuje s databázi MySQL. Dále jsem popsal obsluhu aplikace Hostmin, vyvinutou pro komplexní, ale běžnou správu představených a konfigurovaných balíků.
V důležitých částech jsem zmínil jakým způsobem aplikace Hostmin v pozadí funguje. Analýzou aplikace Hostmin se v této prací nezabývám, protože její rozsah by výrazně navýšil objem této práce a mohl by zastínit důležitější části. Informace uvedené v této práci jsou pouze základním návodem pro tvorbu vlastní webové aplikace. Lze použít různé programovací jazyky a databáze kompatibilní s platformou GNU/Linux. Mnoho z uvedených balíků lze nakonfigurovat za pomocí dodatečných modulu pro spolupráci s jinou databází. Kromě databáze MySQL, jsou to například databáze PostgreSQL nebo Oracle.

Již delší dobu se zajímám o tvorbu webových aplikací nejen v jazyce PHP. Při tvorbě těchto aplikací často využívám hostingových služeb různých společností a v posledních par letech stoupá kvalita služeb. Především jsou častější administrační prostředí, které umožňují měnit různá nastavení služeb nebo spravovat související služby jako je FTP účty, emailové schránky a podobně. Občas jsem se při své práci dostal i k linuxovému serveru a tak jsem se začal zajímat o to, jak takový hostingový server funguje. Nejdříve to začalo konfiguraci serveru bez podpory databáze a při jejich konfiguraci jsem narazil i na možnost využít pro některé balíky modul pro jejich správu s použitím MySQL databáze. Protože mě vytváření webových aplikací baví a začala mě bavit i konfigurace takového serveru pokusil jsem se vytvořit vlastní aplikaci, která by umožnila správu hostingové serveru zjednodušit. V té době se také blížila doba pro vybrání tématu diplomové práce a tak mě napadlo spojit dobré s užitečným a vyvíjet celou aplikaci v rámci diplomové práce.

Aplikace Hostmin je s ohledem na praktičnost a bezpečnost na dobré úrovni a jistě by se našlo její reálné uplatnění a také jak ji dále funkčně rozšířit. Důležité je zdůraznit, že tato práce nestačí jako návod pro běh serveru v reálném provozu. Především zabezpečení serveru ať už z hlediska emailových služeb nebo zabezpečení proti útokům z internetu je nedostačující. Popis instalace a konfigurace veškerého software pouze pro zabezpečení serveru by pravděpodobně přesáhl svým rozsahem celou tuto práci. Avšak kvalitní literatury, návodů a dokumentace je mnoho a co se týká literatury a návodu tak i v českém jazyce. Často se stane, že nic z toho nepomůže a ke správnému zprovoznění je nutné prohledávat různé fóra. Aplikací pro správu serveru lze najít několik univerzálních, ale v poslední době převládá tvorba vlastních. Aplikace je pak přizpůsobena přímo pro konkrétní server, konkrétní konfiguraci a především disponuje příjemným uživatelských rozhraním srozumitelným i pro uživatele laika.

Celá aplikace je na přiloženém CD včetně *.sql souborů pro import do databáze a všech zdrojových souborů.

LITERATURA

[1] NEMETH, Evi – SNYDER, Garth – HEIN, R. Trent. *Linux – Kompletní příručka administrátora, 2. aktualizované vydání.* 1. vyd. Brno: Computer Press, a.s., 2008. 984 s. ISBN 978-80-251-2410-9

[2] HERBORTH, Cris. *Unix a Linux – Názorný průvodce administrátora.* 1. vyd. Brno: Computer Press, a.s., 2006. 288 s. ISBN 80-251-0978-X

[3] HILDEBRANDT, Ralf – KOETTER, Patrick. *Postfix – Provozujeme poštovní server v Linuxu.* 1. vyd. Brno: Computer Press, a.s., 2006. 431 s. ISBN 80-251-1020-6

[4] *Apache HTTP Server Version 2.0 Documentation – Apache HTTP Server*
URL: < http://httpd.apache.org/docs/2.0/> [cit. 2011-01-03]

[5] *The ProFTPD Project: Project Documentation*
URL: < http://www.proftpd.org/docs/> [cit. 2011-04-20]

[6] *Postfix Documentation*
URL: < http://www.proftpd.org/docs/> [cit. 2011-04-20]

[7] *PowerDNS manual*
URL: < http://doc.powerdns.com/> [cit. 2011-04-19]

[8] *HowToForge – Linux Howtos and Tutorials*
URL: < http://howtoforge.com/> [cit. 2011-04-20]

[9] *Debian and disk quotas*
URL:
< http://www.intelligentedu.com/linux_system_administration_course/ch05s02.html>
[cit. 2007-09-11]

[10] *OpenSSL: Documents, Misc:*
URL: < http://www.openssl.org/docs/> [cit. 2009-06-24]